```
ACCESS NAMES TABLE

SOURCE ACCESS NAME=      HC2.LIN.SRC.MORSE
OBJECT ACCESS NAME=      HC2.LIN.OBJ.MORSE
LISTING ACCESS NAME=     HC2.LIN.LST.MORSE
ERROR ACCESS NAME=
OPTIONS=                 XREF
MACRO LIBRARY PATHNAME=
8400  SGCADR EQU  >8400       SOUND CHIP ADDRESS
0012  RSTADR EQU  >0012       POWER UP ENTRY
8C02  VWA    EQU  >8C02       VDP WRITE ADDRESS
FFFE  VWD    EQU  >8C00-VWA   VDP WRITE DATA
FBFE  VRD    EQU  >8800-VWA   VDP READ DATA
4000  WDBIT  EQU  >4000       VDP WRITE DATA CONTROL BIT
0000  RDBIT  EQU  0           VDP READ DATA CONTROL BIT
03C0  SCNMAX EQU  960         TEXT MODE SCREEN SIZE
0600  PGTMAX EQU  >0600       END OF P.G.T., XMIT BUFFER STARTS
0024  COLMAX EQU  36          # OF DISPLAY COLUMNS PER LINE
6000  CHOFFS EQU  >6000       CHAR.OFFSET TO P.G.T.
000E  KEYSCN EQU  >000E       ENTRY POINT OF KEYBOARD SCAN
0019  XWPM   EQU  25          ADDRESS FOR XMIT.WORDS/MIN
0070  XBKMAX EQU  >0070       MAX.OF BLINK TIMER IN XMIT.
0009  XMTLIN EQU  9           LINE # OF XMIT SCREEN
0052  XSCRNO EQU  80+2        TOP OF XMIT SCREEN
0192  XR6MIN EQU  400+2       START OF XMIT SCREEN CURSOR
01B6  XR6MAX EQU  400+38      END OF XMIT SCREEN CURSOR
01F9  RWPM   EQU  480+25      ADDRESS FOR RECV.WORD/MIN.
0700  RBKMAX EQU  >0700       MAX.OF BLINK TIMER IN RECV.
000A  RECLIN EQU  10          LINE # OF RECV.WORDS/MIN
0232  RSCRNO EQU  560+2       TOP OF RECV.SCREEN
039A  RR6MIN EQU  920+2       START OF RECV SCREEN CURSOR
03BE  RR6MAX EQU  920+38      END OF RECV SCREEN CURSOR
0073  RECMAX EQU  >73         SIZE OF RECV.MORSE LOOK-UP TABLE
0500  SHIFTQ EQU  >0500       SHIFT-Q, MASTER RESET
3B00  SHIFTM EQU  >3B00       SHIFT-M, MENU SELECT
0700  SHIFTT EQU  >0700       SHIFT-T, XMIT SELECT
0600  SHIFTR EQU  >0600       SHIFT-R, RECV SELECT
0200  SHIFTC EQU  >0200       SHIFT-C, CONTROL SELECT
0800  BS     EQU  >0800       BACK SPACE
0900  FWD    EQU  >0900       FORWARD KEY
0D00  ENTER  EQU  >0D00       ENTER KEY
8000  ERRORC EQU  >8000       ERROR MORSE CODE
C500  BREAKC EQU  >C500       BREAK MORSE CODE
83E0  PAD    EQU  >83E0       CPU RAM BEGINNING ADDR.
FF20  STOCUR EQU  >8300-PAD   STORAGE FOR CHAR.AT CURSOR
FF22  BUFPTR EQU  >8302-PAD   RECV.BUFFER POINTER
FF24  BUFCTR EQU  >8304-PAD   RECV.BUFFER COUNT
FF26  FLGREG EQU  >8306-PAD   FLAG REGISTER
FF28  BLKTIM EQU  >8308-PAD   CURSOR BLINK TIMER
FF2A  XMTSEL EQU  >830A-PAD   XMIT SPEED-MENU SELECT
FF2C  SAVR12 EQU  >830C-PAD   STORAGE FOR CRU REG.
FF2E  SUBSEC EQU  >830E-PAD   SUB-SECOND COUNT, 250..1
FF30  SEC    EQU  >8310-PAD   SECOND COUNT
FF32  ACCUM  EQU  >8312-PAD   250 HZ ACCUMULATOR
FF34  DDSAV  EQU  >8314-PAD   TWO DOT TIME STORAGE
FF36  CHACNT EQU  >8316-PAD   #OF CHAR.RECEIVED IN 1 MESSAGE
FF38  OLDIN  EQU  >8318-PAD   OLD 250 HZ INPUT LEVEL
FE3A  SAVR5  EQU  >821A-PAD   TEMP.  SAVE R5
FF3C  SAVR6  EQU  >831C-PAD   TEMP.  SAVE R6
FF3E  SBFCTR EQU  >831E-PAD   SAVE LAST RECV.BUFFER COUNT
FF40  RAMBUF EQU  >8320-PAD   CPU BUFFER FOR VDP RAM MOVAL
FF66  OFFSET EQU  >8346-PAD   CHAR. & WORD SPACINGS OFFSET
FF68  WORDC  EQU  >8348-PAD   # OF CONSECUTIVE END OF WORD
FF6A  TWOSPA EQU  >834A-PAD   TWO SPACE TIME
FF6C  TWOEOC EQU  >834C-PAD   TWO END OF CHAR.TIME
FF6E  SAVACC EQU  >834E-PAD   TEMP SAVE 250 HZ ACCUM.
FF70  SUM901 EQU  >8350-PAD   SUM OF 9901 TIMER DATA
FF94  PLAYER EQU  >8374-PAD   PLAYER SELECT CODE FOR KEYSCN
```

```
        FF95   KEY    EQU  >8375-PAD     ASCII CODE RETURNED BY KEYSCN
        FF9C   STATUS EQU  >837C-PAD     STATUS FOR KEYSCN
        FFF4   SAVVDP EQU  >83D4-PAD     VDP REG.1 STORAGE
```

```
0065 4000              RORG >4000
0066 4000   AA         BYTE >AA,1         HEADER, VERSION #
     4001   01
0067 4002   00         BYTE 0,0           # OF PROGRAMS, RESERVED
     4003   00
0068 4004 0000         DATA 0,0           POWER UP, USER PROGRAM HEADERS
     4006 0000
0069 4008 4010'        DATA DSRLNK        DSR HEADER
0070 400A 0000         DATA 0,0           SUBROUTINE, INTERRUPT HEADER
     400C 0000
0071 400E 0000         DATA 0             RESERVED WORD
0072 4010 0000  DSRLNK DATA 0,CWDSR       LINK, DSR ENTRY POINT
     4012 4018'
0073 4014   02         BYTE 2             NAME LENGTH
0074 4015   43         TEXT 'CW'          NAME TEXT
     4016   57
0075 4018              EVEN
0076 4018 02A4  CWDSR  STWP R4            SET UP CPU RAM POINTER
0077 401A 0724         SETO @DDSAV(R4)    INIT.TWODOT TIME TO MAX.
     401C FF34
0078 401E 04E4         CLR  @SBFCTR(R4)   CLR LAST BUFFER COUNT STORAGE
     4020 FF3E
0079 4022 C90C         MOV  R12,@SAVR12(R4)  SAVE CRU REG.
     4024 FF2C
0080 4026 04CC         CLR  R12           DISABLE TIMER INTERRUPT
0081 4028 1E00         SBZ  0
0082 402A 1E03         SBZ  3
0083 402C 06A0  CWDSR1 BL   @BLANKS       BLANK THE SCREEN
     402E 4736'
0084 4030 D7E0         MOVB @HF0,*R15     SET TEXT MODE TO VDP
     4032 4D1A'
0085 4034 D7E0         MOVB @H81,*R15     WRITE INTO VDP REG.1
     4036 4D1C'
0086 4038 D7E0         MOVB @HCE,*R15     SET COLOR GREEN VS.GRAY
     403A 4D24'
0087 403C D7E0         MOVB @H87,*R15     WRITE INTO VDP REG.7
     403E 4D22'
0088 4040 D920         MOVB @H0000,@PLAYER(R4)  PLAYER= CONSOLE CODE
     4042 4D26'
     4044 FF94
0089 4046 D920         MOVB @HF0,@SAVVDP(R4)  SAVE VDP REG.1 DATA
     4048 4D1A'
     404A FFF4
```

```
0091                *
0092                *       *** TONE-MENU SELECTION ***
0093                *
0094 404C 0205  MENUS1 LI   R5,DISPTI      DISPLAY 'TEXAS INSTRUMENTS'
     404E 48FE'
0095 4050 06A0         BL   @MESDSP        * ETC.
     4052 470E'
0096 4054 0205         LI   R5,MENU1       DISPLAY 1ST MENU PAGE
     4056 494A'
0097 4058 06A0         BL   @MESDSP
     405A 470E'
0098 405C 06A0  MENU01 BL   @KEYSCN        GO SCAN KEYBOARD
     405E 000E
0099 4060 02A4         STWP R4             RESTORE R4
0100 4062 D064         MOVB @STATUS(R4),R1  TEST NEW KEY STATUS
     4064 FF9C
0101 4066 13FA         JEQ  MENU01         LOOP IF NO KEY PRESSED
0102 4068 04C1         CLR  R1             CLEAR STORAGE
0103 406A D064         MOVB @KEY(R4),R1    R1= ASCII CODE * 256
     406C FF95
0104 406E 0281         CI   R1,SHIFTQ      MASTER RESET IF SHIFT-Q
     4070 0500
0105 4072 134C         JEQ  RESET
0106 4074 0281         CI   R1,'0'*256     LEGAL SELECTION RANGE: 1..4
     4076 3000
0107 4078 12F1         JLE  MENU01
0108 407A 0281         CI   R1,'4'*256
     407C 3400
0109 407E 15EE         JGT  MENU01
0110 4080 0221         AI   R1,-'1'*256    LET R1= 0, >100, >200 ..
     4082 CF00
0111 4084 0971         SRL  R1,7           NOW R1= 0, 2, 4 ..
0112 4086 C061         MOV  @TONFRQ(R1),R1  R1= FREQ.DATA FOR SOUND CHIP
     4088 4C08'
0113 408A D801         MOVB R1,@SGCADR     SEND FREQ.DATA TO SOUND CHIP'S
     408C 8400
0114 408E 06C1         SWPB R1             * TONE 1 FREQ.REG.
0115 4090 D801         MOVB R1,@SGCADR
     4092 8400
```

```
                *
                *       *** TRANSMITTING SPEED-MENU SELECTION ***
                *
4094 06A0   MENUS2 BL    @BLANKS       BLANK SCREEN
4096 4736'
4098 0205          LI    R5,DISPTI     DISPLAY 'TEXAS INSTRUMENTS'
409A 48FE'
409C 06A0          BL    @MESDSP       * ETC.
409E 470E'
40A0 0205          LI    R5,MENU2      DISPLAY 2ND MENU PAGE
40A2 49C2'
40A4 06A0          BL    @MESDSP
40A6 470E'
40A8 06A0   MENU02 BL    @KEYSCN       GO SCAN KEYBOARD
40AA 000E
40AC 02A4          STWP R4             RESTORE R4
40AE D064          MOVB @STATUS(R4),R1   ANY KEY PRESSED ?
40B0 FF9C
40B2 13FA          JEQ   MENU02        LOOP IF NOT
40B4 04C1          CLR   R1
40B6 D064          MOVB @KEY(R4),R1    R1= ASCII * 256
40B8 FF95
40BA 0281          CI    R1,SHIFTQ     MASTER RESET IF SHIFT-Q
40BC 0500
40BE 1326          JEQ   RESET
40C0 0281          CI    R1,'0'*256    LEGAL SELECTION RANGE: 1..7
40C2 3000
40C4 12F1          JLE   MENU02
40C6 0281          CI    R1,'7'*256
40C8 3700
40CA 15EE          JGT   MENU02
40CC 0221          AI    R1,-'1'*256   LET R1= 0, >100, >200 ..
40CE CF00
40D0 0961          SRL   R1,6          R1= 0, 4, 8 ..
40D2 C901          MOV   R1,@XMTSEL(R4)  SAVE SPEED SELECTION
40D4 FF2A
```

```
                *
                *       *** CONTROL-MENU SELECTION ***
                *
4006 06A0  MENUS3 BL    @BLANKS       BLANK THE SCREEN
4008 4736'
400A D820         MOVB  @DISTON,@SGCADR   DISABLE TONE 1
400C 4D3E'
400E 8400
40E0 C32C         MOV   @SAVR12(R12),R12   RESTORE CRU REG.
40E2 FF2C
40E4 0205         LI    R5,DISPTI     DISPLAY 'TEXAS INSTRUMENTS'
40E6 48FE'
40E8 06A0         BL    @MESDSP       * ETC.
40EA 470E'
40EC 0205         LI    R5,MENU3      DISPLAY 3RD MENU SELECTION
40EE 4AA4'
40F0 06A0         BL    @MESDSP
40F2 470E'
40F4 06A0  MENS03 BL    @KEYSCN       GO SCAN KEYBOARD
40F6 000E
40F8 02A4         STWP  R4            RESTORE R4
40FA D064         MOVB  @STATUS(R4),R1  ANY KEY PRESSED ?
40FC FF9C
40FE 13FA         JEQ   MENS03        LOOP IF NOT
4100 04C1         CLR   R1
4102 D064         MOVB  @KEY(R4),R1   R1= ASCII * 256
4104 FF95
4106 0281         CI    R1,SHIFTQ     RESET  IF SHIFT-Q
4108 0500
410A 1602         JNE   MENS13
410C 0460  RESET  B     @RESETO       MASTER RESET
410E 48DA'
4110 0281  MENS13 CI    R1,SHIFTT     XMIT IF SHIFT-T
4112 0700
4114 1308         JEQ   MENS33
4116 0281         CI    R1,SHIFTR     RECV. IF SHIFT-R
4118 0600
411A 1305         JEQ   MENS33
411C 0281  MENS23 CI    R1,SHIFTM     MENU SELECT IF SHIFT-M
411E 3B00
4120 16E9         JNE   MENS03        LOOP IF OTHER KEYS
4122 0460         B     @CWDSR1
4124 402C'
                *
                *       *** SEPARATE SCREEN IN HALF ***
                *
4126 C181  MENS33 MOV   R1,R6         SAVE SHIFT-T/SHIFT-R CODE
4128 06A0         BL    @BLANKS       BLANK THE SCREEN
412A 4736'
412C 0205         LI    R5,XMTREC     DISPLAY XMIT/RECV SCREEN
412E 4B5C'
4130 06A0         BL    @MESDSP       * SEPARATION
4132 470E'
4134 C064         MOV   @XMTSEL(R4),R1  FROM XMIT SPEED SELECTION,
4136 FF2A
4138 C0A1         MOV   @XMTAB(R1),R2   * FIND SPEED'S ASCII CODE
413A 4C10'
413C 0201         LI    R1,XWPM+WDBIT   SET UP VDP ADDRESS TO
413E 4019
4140 06A0         BL    @VDPWAD         *DISPLAY WORDS/MIN.
```

```
     4142 4704'
0178 4144 0222          AI    R2,CHOFFS     OFFSET TO P.G.T.
     4146 6000
0179 4148 DBC2          MOVB  R2,@VWD(R15)  DISPLAY MSD
     414A FFFE
0180 414C 06C2          SWPB  R2
0181 414E 0222          AI    R2,CHOFFS     OFFSET TO P.G.T.
     4150 6000
0182 4152 DBC2          MOVB  R2,@VWD(R15)  DISPLAY LSD
     4154 FFFE
0183 4156 0286          CI    R6,SHIFTT     XMIT MODE ?
     4158 0700
0184 415A 1302          JEQ   XMIT          GO IF YES
0185 415C 0460          B     @RECV         RECEIVE MODE
     415E 43A2'
```

```
0187                *
0188                *      *** MORSE CODE TRANAMITTER ***
0189                *
0190 4160 04E4  XMIT   CLR  @BUFCTR(R4)   CLR XMIT BUFFER COUNT
     4162 FF24
0191 4164 04E4         CLR  @FLGREG(R4)   CLR FLAG REGISTER
     4166 FF26
0192 4168 04E4         CLR  @BUFPTR(R4)   CLR XMIT BUFFER POINTER
     416A FF22
0193 416C 04E4         CLR  @BLKTIM(R4)   CLR CURSOR BLINK TIMER
     416E FF28
0194 4170 0205         LI   R5,PGTMAX+WDBIT  R5= START OF XMIT BUFFER
     4172 4600
0195 4174 D920         MOVB @BLANKC,@STOCUR(R4)  SET STOCUR= BLANK
     4176 4D18'
     4178 FF20
0196 417A 0206         LI   R6,XR6MIN     R6= XMIT SCREEN CURSOR
     417C 0192
0197 417E C905  XMIT0  MOV  R5,@SAVR5(R4)   SAVE R5
     4180 FE3A
0198 4182 C906         MOV  R6,@SAVR6(R4)   SAVE R6
     4184 FF3C
0199 4186 06A0         BL   @KEYSCN         GO SCAN KEYBOARD
     4188 000E
0200 418A 02A4         STWP R4              RESTORE R4
0201 418C C164         MOV  @SAVR5(R4),R5   RESTORE R5
     418E FE3A
0202 4190 C1A4         MOV  @SAVR6(R4),R6   RESTORE R6
     4192 FF3C
0203 4194 D064         MOVB @STATUS(R4),R1 ANY KEY PRESSED ?
     4196 FF9C
0204 4198 161A         JNE  XMIT4           PROCESS KEY IF YES
0205 419A 05A4         INC  @BLKTIM(R4)   INCR. BLINK TIMER
     419C FF28
0206 419E C064         MOV  @BLKTIM(R4),R1 BLINK TIMER = MAX ?
     41A0 FF28
0207 41A2 0281         CI   R1,XBKMAX
     41A4 0070
0208 41A6 16EB         JNE  XMIT0          GO SCAN KEY IF NOT
0209 41A8 04E4         CLR  @BLKTIM(R4)    CLR BLINK TIMER
     41AA FF28
0210 41AC C046         MOV  R6,R1          R1= CURSOR ADDRESS
0211 41AE 0221         AI   R1,WDBIT       ADD  CONTROL BIT
     41B0 4000
0212 41B2 06A0         BL   @VDPWAD        SET UP VDP ADDRESS
     41B4 4704'
0213 41B6 B920         AB   @H80,@FLGREG(R4) XOR BLINK FLAG ( MSB )
     41B8 4D2A'
     41BA FF26
0214 41BC 1104         JLT  XMIT2          GO IF MSB SET
0215 41BE DBE4         MOVB @STOCUR(R4),@VWD(R15)  DISPLAY STORED CHAR.
     41C0 FF20
     41C2 FFFE
0216 41C4 10DC         JMP  XMIT0
0217 41C6 DBE0  XMIT2  MOVB @CURSOR,@VWD(R15)  DISPLAY CURSOR CHAR.
     41C8 4D28'
     41CA FFFE
0218 41CC 10D8  XMIT00 JMP  XMIT0          GO SCAN KEYBOARD
0219 41CE 04C2  XMIT4  CLR  R2
0220 41D0 D0A4         MOVB @KEY(R4),R2   R2= ASCII CODE * 256
```

```
     41D2 FF95
0221 41D4 0282         CI    R2,SHIFTC     KEY= SHIFT C ?
     41D6 0200
0222 41D8 1602         JNE   XMIT3         GO IF NOT
0223 41DA 0460         B     @MENUS3       PROCESS CONTROL SELECT
     41DC 40D6'
0224 41DE C046  XMIT3  MOV   R6,R1         ACCESS CURSOR
0225 41E0 0221         AI    R1,WDBIT
     41E2 4000
0226 41E4 06A0         BL    @VDPWAD                  PUT ORIGINAL
     41E6 4704'
0227 41E8 DBE4         MOVB  @STOCUR(R4),@VWD(R15)  * CHAR.BACK
     41EA FF20
     41EC FFFE
0228 41EE C042         MOV   R2,R1         R1= NEW KEY
0229 41F0 0281         CI    R1,ENTER      KEY= ENTER ?
     41F2 0D00
0230 41F4 1602         JNE   XMIT5         GO IF NOT
0231 41F6 0460         B     @XMTEN        START  OF XMISSION
     41F8 42BE'
0232 41FA 0281  XMIT5  CI    R1,BS         KEY= BACK SPACE ?
     41FC 0800
0233 41FE 134B         JEQ   XMTBS         GO IF YES
0234 4200 0281         CI    R1,FWD        KEY= FORWARD ?
     4202 0900
0235 4204 1308         JEQ   XMTFWD        GO IF YES
0236 4206 0281         CI    R1,' '*256    ILLEGAL KEY IF < ' '
     4208 2000
0237 420A 11E0         JLT   XMIT00
0238 420C 0281         CI    R1,'Z'*256    ILLEGAL KEY IF > 'Z'
     420E 5A00
0239 4210 15DD         JGT   XMIT00
0240 4212 C081         MOV   R1,R2         R2= NEW CHAR.
0241 4214 1005         JMP   XMIT6
0242 4216 04C2  XMTFWD CLR   R2               R2= STORED CHAR.AT CURSOR
0243 4218 D0A4         MOVB  @STOCUR(R4),R2
     421A FF20
0244 421C 0222         AI    R2,-CHOFFS    GET RID OF OFFSET TO P.G.T.
     421E A000
0245 4220 C002  XMIT6  MOV   R2,R0         R0= TEMP.STORAGE FOR CHAR.
0246 4222 0222         AI    R2,-' '*256     MINUS SPACE CODE THEN SHIFT
     4224 E000
0247 4226 0972         SRL   R2,7            *RIGHT 7 FOR INDEX INTO XMIT
0248 4228 C062         MOV   @XMTDD(R2),R1 *TABLE
     422A 4C2C'
0249 422C 13CF         JEQ   XMIT00        ILLEGAL IF DATA IN TABLE= 0
0250 422E 0220         AI    R0,CHOFFS     ADD OFFSET TO P.G.T.
     4230 6000
0251 4232 C046         MOV   R6,R1         ACCESS CURSOR ADDRESS
0252 4234 0221         AI    R1,WDBIT      ADD CONTROL BIT
     4236 4000
0253 4238 06A0         BL    @VDPWAD
     423A 4704'
0254 423C DBC0         MOVB  R0,@VWD(R15)  DISPLAY CHAR.ON CURSOR
     423E FFFE
0255 4240 C045         MOV   R5,R1         ACCESS XMIT BUFFER IN VDP
0256 4242 06A0         BL    @VDPWAD
     4244 4704'
0257 4246 06C2         SWPB  R2            STORE INDEX OF XMIT TABLE
0258 4248 DBC2         MOVB  R2,@VWD(R15)  *IN XMIT BUFFER
```

```
     424A FFFE
0259 424C 05A4         INC   @BUFPTR(R4)   INCR. BUFFER POINTER
     424E FF22
0260 4250 0585         INC   R5            NEXT BYTE IN XMIT BUFFER
0261 4252 0586         INC   R6            INCR. CURSOR ADDRESS
0262 4254 8924         C     @BUFPTR(R4),@BUFCTR(R4)  POINTER > COUNT ?
     4256 FF22
     4258 FF24
0263 425A 150A         JGT   XMIT7         GO IF YES
0264 425C 04C1         CLR   R1            SET R1 & R1 = 0 & CURSOR
0265 425E C086         MOV   R6,R2
0266 4260 3C60         DIV   @D40,R1       CURSOR/40
     4262 4D2C'
0267 4264 0282         CI    R2,38         REMAINDER= 38 ?
     4266 0026
0268 4268 160E         JNE   XMIT8         GO IF NOT
0269 426A 0226         AI    R6,4          END OF LINE, ADD 4 TO CURSOR
     426C 0004
0270 426E 100B         JMP   XMIT8
0271 4270 05A4  XMIT7  INC   @BUFCTR(R4)   INCR. BUFFER COUNT
     4272 FF24
0272 4274 0286         CI    R6,XR6MAX     CURSOR= MAX ?
     4276 01B6
0273 4278 1606         JNE   XMIT8         GO IF NOT
0274 427A 0200         LI    R0,XMTLIN-1   R0= # OF LINES TO SCROLL
     427C 0008
0275 427E 0201         LI    R1,XSCRNO     R1= TOP OF XMIT SCREEN
     4280 0052
0276 4282 06A0         BL    @SCROLL       GO SCROLL XMIT SCREEN
     4284 4754'
0277 4286 C046  XMIT8  MOV   R6,R1         READ CHAR. AT CURSOR
0278 4288 06A0         BL    @VDPWAD
     428A 4704'
0279 428C D92F         MOVB  @VRD(R15),@STOCUR(R4)
     428E FBFE
     4290 FF20
0280 4292 0460         B     @XMITO        GO SCAN NEXT KEY
     4294 417E'
0281 4296 C024  XMTBS  MOV   @BUFPTR(R4),R0  BUFFER POINTER= 0 ?
     4298 FF22
0282 429A 13F5         JEQ   XMIT8         GO IF YES
0283 429C 0286         CI    R6,XSCRNO     CURSOR= TOP OF XMIT SCREEN ?
     429E 0052
0284 42A0 13F2         JEQ   XMIT8         GO IF YES
0285 42A2 0624         DEC   @BUFPTR(R4)   DECR. BUFFER POINTER
     42A4 FF22
0286 42A6 0605         DEC   R5            BACK 1 BYTE IN XMIT BUFFER
0287 42A8 0606         DEC   R6            DECR. CURSOR
0288 42AA 04C1         CLR   R1            SET R1 & R2 = 0 & CURSOR
0289 42AC C086         MOV   R6,R2
0290 42AE 3C60         DIV   @D40,R1       CURSOR/40
     42B0 4D2C'
0291 42B2 0282         CI    R2,1          REMAINDER= 1 ?
     42B4 0001
0292 42B6 16E7         JNE   XMIT8         GO IF NOT
0293 42B8 0226         AI    R6,-4         TOP OF LINE, CURSOR-4
     42BA FFFC
0294 42BC 10E4         JMP   XMIT8         STORE CHAR. AT CURSOR
```

```
0296                *
0297                *       *** END OF EDIT, START OF XMISSION ***
0298                *
0299 42BE C064      XMTEN   MOV   @XMTSEL(R4),R1  R1= 0, 2, 4, 8 ..
     42C0 FF2A
0300 42C2 05C1              INCT R1              R1= 2, 4, 6, 10 ..
0301 42C4 C021              MOV   @XMTAB(R1),R0 R0= 1 DOT TIME
     42C6 4C10'
0302 42C8 04E4              CLR   @BUFPTR(R4)    RESET  BUFFER POINTER
     42CA FF22
0303 42CC C324              MOV   @SAVR12(R4),R12  RESTORE R12
     42CE FF2C
0304 42D0 C064              MOV   @BUFCTR(R4),R1  NO CHAR.TO SEND ?
     42D2 FF24
0305 42D4 1603              JNE   XMTEN8        XMIT THIS MESSAGE IF NOT
0306 42D6 C924              MOV   @SBFCTR(R4),@BUFCTR(R4)  XMIT LAST MESS.IF Y
     42D8 FF3E
     42DA FF24
0307 42DC 0201      XMTEN8 LI    R1,PGTMAX     R1= START OF XMIT BUFFER
     42DE 0600
0308 42F0 06A0              BL    @VDPWAD
     42F2 4704'
0309 42F4 8924      XMTEN1 C     @BUFPTR(R4),@BUFCTR(R4) ALL CHAR.XMITTED ?
     42F6 FF22
     42F8 FF24
0310 42FA 1607              JNE   XMTEN2
0311 42FC 0201              LI    R1,10         DELAY 10 DOT TIME
     42FE 000A
0312 42F0 06A0              BL    @DELAY
     42F2 4372'
0313 42F4 C160              MOV   @EOM,R5       R5= END OF MESSAGE
     42F6 4CA2'
0314 42F8 1005              JMP   XMTEN3
0315 42FA D16F      XMTEN2 MOVB  @VRD(R15),R5 FETCH INDEX STORED IN BUFFER
     42FC FBFE
0316 42FE 0985              SRL   R5,8          PUT TO LOWER BYTE
0317 4300 C165              MOV   @XMTDD(R5),R5  R5= XMIT DOT-DASH DATA
     4302 4C2C'
0318 4304 C0C5      XMTEN3 MOV   R5,R3
0319 4306 0983              SRL   R3,8          R3= BIT #
0320 4308 06C5              SWPB R5             HIGH BYTE R5= DOT-DASH
0321 430A 0283              CI    R3,>FF        SPACE CODE ?
     430C 00FF
0322 430E 1605              JNE   XMTEN4        GO IF NOT
0323 4310 0201              LI    R1,4          TIME UNIT= 4
     4312 0004
0324 4314 06A0              BL    @DELAY        DELAY 4 DOT (END OF WORD)
     4316 4372'
0325 4318 101B              JMP   XMTEN7
0326 431A 0A15      XMTEN4 SLA   R5,1          MSB SET ?
0327 431C 1803              JOC   XMTEN5        GO IF YES
0328 431E 0201              LI    R1,1          TIME UNIT= 1 (DOT)
     4320 0001
0329 4322 1002              JMP   XMTEN6
0330 4324 0201      XMTEN5 LI    R1,3          TIME UNIT= 3 (DASH)
     4326 0003
0331 4328 D820      XMTEN6 MOVB  @ENTONE,@SGCADR  ENABLE TONE 1
     432A 4D3C'
     432C 8400
0332 432E 1D01              SBO   1             SET XMIT LINE 1
```

```
0333 4330 06A0         BL    @DELAY          DELAY DOT/DASH TIME
     4332 4372'
0334 4334 1E01         SBZ   1               SET  XMIT LINE 0
0335 4336 D820         MOVB  @DISTON,@SGCADR  DISABLE TONE 1
     4338 4D3E'
     433A 8400
0336 433C 0201         LI    R1,1            TIME UNIT= 1 FOR SPACING
     433E 0001
0337 4340 06A0         BL    @DELAY          DELAY 1 DOT TIME
     4342 4372'
0338 4344 0603         DEC   R3              DECR.BIT #
0339 4346 16E9         JNE   XMTEN4          BACK IF NOT DONE
0340 4348 0201         LI    R1,2            TIME UNIT= 2
     434A 0002
0341 434C 06A0         BL    @DELAY          END CHAR.DELAY (2 DOT TIME)
     434E 4372'
0342 4350 05A4  XMTEN7 INC   @BUFPTR(R4)     INCR.BUFFER POINTER
     4352 FF22
0343 4354 8924         C     @BUFPTR(R4),@BUFCTR(R4)  PTR. > COUNT ?
     4356 FF22
     4358 FF24
0344 435A 12C4         JLE   XMTEN1          GO IF NOT
0345 435C 0200         LI    R0,XMTLIN-1     R0= # OF LINES TO SCROLL
     435E 0008
0346 4360 0201         LI    R1,XSCRN0       R1= TOP OF XMIT SCREEN
     4362 0052
0347 4364 06A0         BL    @SCROLL         GO SCROLL XMIT SCREEN
     4366 4754'
0348 4368 C924         MOV   @BUFCTR(R4),@SBFCTR(R4)  STORE BUFFER COUNT
     436A FF24
     436C FF3E
0349 436E 0460         B     @XMIT           GOTO START OF XMIT
     4370 4160'
```

```
0351                *
0352                *      *** DELAY SEVERAL UNITS (SPECIFIED BY R1) OF
0353                *      *** TIME, WHICH IS BASED ON R0
0354                *
0355 4372 04CC  DELAY   CLR   R12            SET TO 9901 ADDRESS .
0356 4374 33E0         LDCR  @MAXTIM,15     INIT.TIMER TO MAX.
     4376 4D40'
0357 4378 1E00         SBZ   0              FREE CLOCK BUFFER FOR UPDATE
0358 437A 04CA         CLR   R10            CLR STORAGE
0359 437C 020C         LI    R12,2
     437E 0002
0360 4380 1000  DELAY1 NOP                  DELAY AT LEAST 1 CLOCK CYCLE
0361 4382 1000         NOP
0362 4384 1000         NOP
0363 4386 1000         NOP
0364 4388 1000         NOP
0365 438A 1000         NOP
0366 438C 1000         NOP
0367 438E 1DFF         SBO   -1             DISABLE CLOCK BUFFER FOR UPDATE
0368 4390 378A         STCR  R10,14         R10= 9901 TIMER
0369 4392 1EFF         SBZ   -1             FREE  CLOCK BUFFER FOR UPDATE
0370 4394 800A         C     R10,R0         TIMER DOWN TO DESIRED VALUE ?
0371 4396 15F4         JGT   DELAY1         LOOP IF NOT
0372 4398 0601         DEC   R1             DECR.TIME UNIT
0373 439A 16EB         JNE   DELAY          BACK IF NOT ALL DONE
0374 439C C324         MOV   @SAVR12(R4),R12  RESTORE CRU REG.
     439E FF2C
0375 43A0 045B         RT
```

```
0377                *
0378                *          *** MORSE CODE RECEIVER ***
0379                *
0380 43A2 04E4 RECV    CLR   @FLGREG(R4)   CLR FLAGS
     43A4 FF26
0381 43A6 04E4         CLR   @WORDC(R4)    CLR WORD COUNT
     43A8 FF68
0382 43AA C920         MOV   @D300,@TWOSPA(R4)   INIT TWOSPA TO 300 M.SEC.
     43AC 4D38'
     43AE FF6A
0383 43B0 C920         MOV   @D900,@TWOEOC(R4)   INIT TWOEOC TO 900 M.SEC.
     43B2 4D36'
     43B4 FF6C
0384 43B6 0207         LI    R7,1          INIT.RECV.REG.TO 1
     43B8 0001
0385 43BA 0206         LI    R6,RR6MIN     R6= RECV.SCREEN CURSOR
     43BC 039A
0386 43BE C024         MOV   @DDSAV(R4),R0   R0= TWODOT TIME
     43C0 FF34
0387 43C2 04E4         CLR   @SEC(R4)      CLEAR SECONR COUNT
     43C4 FF30
0388 43C6 C920         MOV   @D250,@SUBSEC(R4)   INIT.SUBSECOND COUNT
     43C8 4D3A'
     43CA FF2E
0389 43CC 04E4         CLR   @BLKTIM(R4)   CLR CURSOR BLINK TIMER
     43CE FF28
0390 43D0 D920         MOVB  @BLANKC,@STOCUR(R4)   INIT.CURSOR STORAGE
     43D2 4D18'
     43D4 FF20
0391 43D6 04E4         CLR   @CHACNT(R4)   CLR RECEIVED CHAR.COUNT
     43D8 FF36
0392 43DA C324 RECV0   MOV   @SAVR12(R4),R12   RESTORE CRU REG.
     43DC FF2C
0393 43DE 1F00 RECV00  TB    0             KEYER DOWN ?
0394 43E0 1326         JEQ   KEYDWN        GO IF YES
0395 43E2 D820 RECV4   MOVB  @DISTON,@SGCADR   DISABLE TONE 1
     43E4 4D3E'
     43E6 8400
0396 43E8 06A0         BL    @SHC          TEST SHIFT-C
     43EA 4800'
0397 43EC 1302         JEQ   RECV2         CONTINUE IF NOT SHIFT-C
0398 43EE 0460         B     @MENUS3       CONTROL SELECT
     43F0 40D6'
0399 43F2 C324 RECV2   MOV   @SAVR12(R4),R12   RESTORE CRU REG.
     43F4 FF2C
0400 43F6 06A0         BL    @HZ250        CHECK OSCILLATOR IN
     43F8 4814'
0401 43FA 05A4         INC   @BLKTIM(R4)   INCR.BLINK TIMER
     43FC FF28
0402 43FE C064         MOV   @BLKTIM(R4),R1
     4400 FF28
0403 4402 0281         CI    R1,RBKMAX     BLINK TIMER= MAX ?
     4404 0700
0404 4406 16EB         JNE   RECV00        GO IF NOT
0405 4408 04E4         CLR   @BLKTIM(R4)   RESET BLINK TIMER
     440A FF28
0406 440C C046         MOV   R6,R1         ACCESS CURSOR IN VDP
0407 440E 0221         AI    R1,WDBIT
     4410 4000
0408 4412 06A0         BL    @VDPWAD
```

```
     4414 4704'
0409 4416 B920         AB    @H80,@FLGREG(R4)  XOR BLINK TIMER
     4418 4D2A'
     441A FF26
0410 441C 1104         JLT   RECV3         GO IF FLAG SET
0411 441E DBE4         MOVB  @STOCUR(R4),@VWD(R15)  DISPLAY STORED CHAR.
     4420 FF20
     4422 FFFE
0412 4424 10DC         JMP   RECV00        CHECK KEYER
0413 4426 DBE0  RECV3  MOVB  @CURSOR,@VWD(R15)  DISPLAY CURSOR CHAR.
     4428 4D28'
     442A FFFE
0414 442C 10D8         JMP   RECV00
0415 442E 04CC  KEYDWN CLR   R12           ACCESS 9901 TIMER
0416 4430 33E0         LDCR  @MAXTIM,15    RESET 9901 TIMER TO MAX
     4432 4D40'
0417 4434 1E00         SBZ   0             FREE CLOCK BUFFER FOR UPDATE
0418 4436 04E4         CLR   @ACCUM(R4)    CLR 250 HZ ACCUM.
     4438 FF32
0419 443A D820         MOVB  @ENTONE,@SGCADR  ENABLE TONE 1
     443C 4D3C'
     443E 8400
0420 4440 C046         MOV   R6,R1         ACCESS CURSOR
0421 4442 0221         AI    R1,WDBIT      ADD CONTROL BIT
     4444 4000
0422 4446 06A0         BL    @VDPWAD
     4448 4704'
0423 444A DBE0         MOVB  @BLANKC,@VWD(R15)  BLANK CURSOR
     444C 4D18'
     444E FFFE
0424 4450 C324         MOV   @SAVR12(R4),R12  RESTORE R12
     4452 FF2C
0425 4454 06A0         BL    @DEBOUN       DEBOUNCE DELAY
     4456 487E'
0426 4458 16C4         JNE   RECV4         GO IF KEYER STILL UP
0427 445A F920         SOCB  @RTFLAG,@FLGREG(R4)  SET REALTIME FLAG
     445C 4D46'
     445E FF26
0428 4460 04E4         CLR   @SUM901(R4)   CLR SUM OF 9901 TIMER DATA
     4462 FF70
0429 4464 1F00  KEYDW0 TB    0             KEYER UP ?
0430 4466 160A         JNE   KEYUP         GO IF YES
0431 4468 06A0  KEYDW2 BL    @SHC          SHIFT-C PRESSED ?
     446A 4800'
0432 446C 1302         JEQ   KEYDW1        GO IF NOT
0433 446E 0460         B     @MENUS3       CONTROL SELECTION
     4470 40D6'
0434 4472 C324  KEYDW1 MOV   @SAVR12(R4),R12  RESTORE CRU REG.
     4474 FF2C
0435 4476 06A0         BL    @HZ250        CHECK OSCILLATOR IN
     4478 4814'
0436 447A 10F4         JMP   KEYDW0
0437 447C 020C  KEYUP  LI    R12,2         ACCESS 9901 CLOCK
     447E 0002
0438 4480 04CA         CLR   R10           CLR STORAGE
0439 4482 1DFF         SBO   -1            DISABLE CLOCK BUFFER FOR UPDATE
0440 4484 378A         STCR  R10,14        R10= 9901 TIMER
0441 4486 04CC  KEYUP1 CLR   R12
0442 4488 33E0         LDCR  @MAXTIM,15    SET 9901 TIMER TO MAX
     448A 4D40'
```

```
0443 448C 1E00          SBZ  0              FREE CLOCK BUFFER FOR UPDATE
0444 448E D820          MOVB @DISTON,@SGCADR  DISABLE TONE 1
     4490 4D3E'
     4492 8400
0445 4494 0209          LI   R9,>3FFF       ADJUST TIMER DATA TO COUNT UP
     4496 3FFF
0446 4498 624A          S    R10,R9
0447 449A A909          A    R9,@SUM901(R4)  PUT NEW TIMER DATA IN SUM
     449C FF70
0448 449E 06A0          BL   @DEBOUN        DEBOUNCE DELAY
     44A0 487E'
0449 44A2 1604          JNE  KEYUP2         GO IF KEYER STILL UP
0450 44A4 D820          MOVB @ENTONE,@SGCADR   ENABLE TONE 1
     44A6 4D3C'
     44A8 8400
0451 44AA 10DE          JMP  KEYDW2         IGNORE SIGNAL UP GLICH
0452 44AC 06A0   KEYUP2 BL   @SCALE         SCALE R8/R10 INTO R1
     44AE 4854'
0453 44B0 8001          C    R1,R0          DOWN TIME < TWODOT ?
0454 44B2 1A0A          JL   KDOT           GO IF NOT
0455             *
0456 44B4 0A17   KDASH  SLA  R7,1           RECV REG. * 2
0457 44B6 0587          INC  R7             INCR RECV.REG.
0458 44B8 C0C0          MOV  R0,R3          R3= TWODOT
0459 44BA 0913          SRL  R3,1           TWODOT/2
0460 44BC 6043          S    R3,R1          THREE DOTS - TWODOT/2
0461 44BE C080          MOV  R0,R2          R2= TWODOT
0462 44C0 6042          S    R2,R1          - OLD TWODOT + NEW TWODOT
0463 44C2 0821          SRA  R1,2           DIFFERENCE / 4
0464 44C4 A001          A    R1,R0          UPDATE OLD TWO DOT
0465 44C6 1006          JMP  KDD
0466             *
0467 44C8 0A17   KDOT   SLA  R7,1           RECV.REG. * 2
0468 44CA 0A11          SLA  R1,1           DOWN TIME * 2
0469 44CC C080          MOV  R0,R2          R2= TWODOT
0470 44CE 6042          S    R2,R1          - TWODOT + DOWNTIME * 2
0471 44D0 0821          SRA  R1,2           DIFFERENCE / 4
0472 44D2 A001          A    R1,R0          UPDATE TWODOT TIME
0473 44D4 06A0   KDD    BL   @CHWRD       UPDATE CHAR.&WORD SPACING
     44D6 48A2'
0474 44D8 C324          MOV  @SAVR12(R4),R12  RESTORE CRU REG.
     44DA FF2C
0475 44DC 1F00   KDDO   TB   0              KEYER DOWN ?
0476 44DE 130F          JEQ  KEYDN
0477 44E0 06A0   KDD1   BL   @HZ250         CHECK OSCILLATOR IN
     44E2 4814'
0478 44E4 C064          MOV  @TWOEOC(R4),R1  R1= TWO E.O.CHAR.PER M.SEC.
     44E6 FF6C
0479 44E8 0911          SRL  R1,1           R1= 4 *  E.O.C./ 4 M.SEC.
0480 44EA 8064          C    @ACCUM(R4),R1  ACCUM. < 4 E.O.CHAR.TIME ?
     44EC FF32
0481 44EE 11F6          JLT  KDDO           LOOP  IF YES
0482 44F0 F920          SOCB @WORD,@FLGREG(R4)  SET WORD FLAG
     44F2 4D4A'
     44F4 FF26
0483 44F6 F920          SOCB @FORCE,@FLGREG(R4) SET FORCED FLAG
     44F8 4D48'
     44FA FF26
0484 44FC 104F          JMP  DECODE         DECODE RECV.REG.
0485             *
```

```
0486 44FE 020C  KEYDN   LI    R12,2          ACCESS 9901 TIMER
     4500 0002
0487 4502 04CA          CLR   R10
0488 4504 1DFF          SBO   -1             DISABLE CLOCK BUFFER FOR UPDATE
0489 4506 378A          STCR  R10,14         READ TIMER INTO R10
0490 4508 04CC          CLR   R12
0491 450A 33E0          LDCR  @MAXTIM,15     RESET 9901 TIMER TO MAX
     450C 4D40'
0492 450E 1E00          SBZ   0              FREE CLOCK BUFFER FOR UPDATE
0493 4510 D820          MOVB  @ENTONE,@SGCADR  ENABLE TONE 1
     4512 4D3C'
     4514 8400
0494 4516 0209          LI    R9,>3FFF       ADJUST TIMER DATA TO COUNT UP
     4518 3FFF
0495 451A 624A          S     R10,R9
0496 451C A909          A     R9,@SUM901(R4)  PUT NEW TIMER DATA IN SUM
     451E FF70
0497 4520 06A0          BL    @DEBOUN        DEBOUNCE DELAY
     4522 487E'
0498 4524 1304          JEQ   KEYDN1         GO IF KEYER STILL DOWN
0499 4526 D820          MOVB  @DISTON,@SGCADR  DISABLE TONE 1
     4528 4D3E'
     452A 8400
0500 452C 10D9          JMP   KDD1           IGNORE SIGNAL DOWN GLICH
0501            *
0502 452E 06A0  KEYDN1  BL    @SCALE         SCALE UPTIME INTO R1
     4530 4854'
0503 4532 8901          C     R1,@TWOSPA(R4)  UPTIME > TWO SPACE ?
     4534 FF6A
0504 4536 1B0B          JH    KEYDN0         GO IF YES
0505 4538 0A11          SLA   R1,1           UPTIME * 2
0506 453A C0A4          MOV   @TWOSPA(R4),R2  R2= TWO SPACE
     453C FF6A
0507 453E 6042          S     R2,R1          - TWO SPACE + 2 * UPTIME
0508 4540 0821          SRA   R1,2           DIFFERENCE / 4
0509 4542 A901          A     R1,@TWOSPA(R4)  UPDATE TWO SPACE
     4544 FF6A
0510 4546 04E4          CLR   @WORDC(R4)     CLR WORD COUNT
     4548 FF68
0511 454A 0460          B     @RECV0         SPACE IS BETWEEN DOT/DASH
     454C 43DA'
0512 454E C0A4  KEYDN0  MOV   @TWOEOC(R4),R2  R2= TWO E.O.CHAR.
     4550 FF6C
0513 4552 A0A4          A     @OFFSET(R4),R2  ADJUST FOR CHAR. & WORD
     4554 FF66
0514 4556 8081          C     R1,R2          UPTIME > TWO END OF CHAR. ?
0515 4558 1B15          JH    KEYDN2         END OF WORD IF YES
0516 455A 04E4          CLR   @WORDC(R4)     CLR WORD COUNT
     455C FF68
0517 455E C0A4          MOV   @TWOEOC(R4),R2  R2= TWO END OF CHAR.
     4560 FF6C
0518 4562 0A11          SLA   R1,1           UPTIME * 2
0519 4564 6042          S     R2,R1          - TWO E.O.CHAR. + 2 * UPTIME
0520 4566 0821          SRA   R1,2           DIFFERENCE / 4
0521 4568 A901          A     R1,@TWOEOC(R4)  ADJUST TWO E.O.CHAR.
     456A FF6C
0522 456C C0A4          MOV   @TWOEOC(R4),R2  R2= NEW TWOEOC
     456E FF6C
0523 4570 0912          SRL   R2,1
0524 4572 C0C2          MOV   R2,R3
```

```
0525 4574 0922         SRL  R2,2
0526 4576 A083         A    R3,R2         R2= 5 * TWOEOC / 8
0527 4578 8902         C    R2,@TWOSPA(R4)  5*TWOEOC/8 > TWO SPACE ?
     457A FF6A
0528 457C 1B0F         JH   DECODE        O.K. IF HIGHER
0529 457E C902         MOV  R2,@TWOSPA(R4)  FORCE TWOSPA TO LOWER VALUE
     4580 FF6A
0530 4582 100C         JMP  DECODE        DECODE THE DATA
0531 4584 F920  KEYDN2 SOCB @WORD,@FLGREG(R4)  SET WORD FLAG (END WORD)
     4586 4D4A'
     4588 FF26
0532 458A 05A4         INC  @WORDC(R4)    INCR WORD COUNT
     458C FF68
0533 458E 8824         C    @WORDC(R4),@D5  IF 5 CONSECUTIVE E.O.WORD,
     4590 FF68
     4592 4D34'
0534 4594 1103         JLT  DECODE            * FORCE E.O.CHAR. TO LARGER
0535 4596 0A11         SLA  R1,1              * VALUE
0536 4598 C901         MOV  R1,@TWOEOC(R4)
     459A FF6C
```

```
0538 459C C900  DECODE MOV  R0,@DDSAV(R4)  SAVE TWODOT TIME
     459E FF34
0539 45A0 C324         MOV  @SAVR12(R4),R12  RESTORE  R12
     45A2 FF2C
0540 45A4 06A0         BL   @HZ250          CHECK OSCILLATOR IN
     45A6 4814'
0541 45A8 06C7         SWPB R7              GET DATA IN HIGH BYTE
0542 45AA D1C7         MOVB R7,R7           CHECK RECV.CODE
0543 45AC 1156         JLT  DECOD4          GO IF MSB(DATA) SET
0544 45AE 1358         JEQ  DECSP1          GO IF DATA= 0
0545 45B0 06C7         SWPB R7              PUT DATA IN LOW BYTE
0546 45B2 0287         CI   R7,RECMAX       RECV.CODE > SIZE OF LOOK-UP ?
     45B4 0073
0547 45B6 154C         JGT  DECODX          ILLEGAL CHAR.IF YES
0548 45B8 04C2         CLR  R2
0549 45BA D0A7         MOVB @RECTAB(R7),R2  R2= ASCII OF RECV.CHAR.
     45BC 4CA4'
0550 45BE 1348         JEQ  DECODX          ILLEGAL CHAR.IF DATA= 0
0551 45C0 0282         CI   R2,' '*256      SPECIAL CHAR.IF < ' '
     45C2 2000
0552 45C4 1150         JLT  DECSP
0553 45C6 05A4         INC  @CHACNT(R4)     INCR.CHAR.COUNT
     45C8 FF36
0554 45CA 0222         AI   R2,CHOFFS       ADD OFFSET TO P.G.T.
     45CC 6000
0555 45CE C046  DECOD8 MOV  R6,R1           ACCESS CURSOR
0556 45D0 0221         AI   R1,WDBIT
     45D2 4000
0557 45D4 06A0         BL   @VDPWAD
     45D6 4704'
0558 45D8 DBC2         MOVB R2,@VWD(R15) DISPLAY CHAR.ON SCREEN
     45DA FFFE
0559 45DC 0586         INC  R6              INCR.CURSOR
0560 45DE 0286  DECOD3 CI   R6,RR6MAX       CURSOR= MAX ?
     45E0 03BE
0561 45E2 1610         JNE  DECOD1
0562 45E4 0200         LI   R0,RECLIN-1     R0= # OF LINES TO SCROLL
     45E6 0009
0563 45E8 0201         LI   R1,RSCRNO       R1= TOP OF RECV.SCREEN
     45EA 0232
0564 45EC 06A0         BL   @SCROLL         GO SCROLL RECV.SCREEN
     45EE 4754'
0565 45F0 D224         MOVB @FLGREG(R4),R8  GET FLAG REG.
     45F2 FF26
0566 45F4 2220         COC  @WORD,R8        WORD FLAG SET ?
     45F6 4D4A'
0567 45F8 160F         JNE  DECOD2          GO IF NOT
0568 45FA 0586         INC  R6              INCR.CURSOR
0569 45FC 5920  DECOD7 SZCB @WORD,@FLGREG(R4)  CLR WORD FLAG
     45FE 4D4A'
     4600 FF26
0570 4602 100A         JMP  DECOD2
0571 4604 D224  DECOD1 MOVB @FLGREG(R4),R8  GET FLAG REG.
     4606 FF26
0572 4608 2220         COC  @WORD,R8        WORD FLAG SET ?
     460A 4D4A'
0573 460C 1605         JNE  DECOD2          GO IF NOT
0574 460E 5920         SZCB @WORD,@FLGREG(R4)  CLR WORD FLAG
     4610 4D4A'
     4612 FF26
```

```
0575 4614 0586          INC  R6             INCR CURSOR
0576 4616 10E3          JMP  DECOD3
0577               *
0578               *    *** ALL DECODE PROCESS EXIT HERE
0579               *
0580 4618 0207     DECOD2 LI   R7,1           INIT.RECV.REG.TO 1
     461A 0001
0581 461C C024          MOV  @DDSAV(R4),R0  RESTORE TWODOT TIME
     461E FF34
0582 4620 D224          MOVB @FLGREG(R4),R8  GET FLAG REG.
     4622 FF26
0583 4624 2220          COC  @FORCE,R8      FORCED FLAG ON ?
     4626 4D48'
0584 4628 130B          JEQ  DECOD5         GO IF YES
0585 462A 2220          COC  @SEMIFR,R8     SEMI-FORCED FLAG ON ?
     462C 4D44'
0586 462E 1606          JNE  DECO10         GO IF NOT
0587 4630 5920          SZCB @SEMIFR,FLGREG(R4)  CLR SEMI-FORCED FLAG
     4632 4D44'
     4634 FF26
0588 4636 06A0          BL   @DEBOUN        DEBOUNCE DELAY
     4638 487E'
0589 463A 1602          JNE  DECOD5         GO IF KEYER STILL UP
0590 463C 0460     DECO10 B    @KEYDWO        CHECK IF KEYER IS UP
     463E 4464'
0591 4640 5920     DECOD5 SZCB @FORCE,@FLGREG(R4)  CLR FORCED FLAG
     4642 4D48'
     4644 FF26
0592 4646 D820          MOVB @DISTON,@SGCADR  DISABLE TONE 1
     4648 4D3E'
     464A 8400
0593 464C 0460          B    @RECV00        CHECK IF KEYER IS DOWN
     464E 43DE'
0594               *
0595 4650 0202     DECODX LI   R2,'^'*256     DISPLAY '^' TO INDICATE
     4652 5E00
0596 4654 0222          AI   R2,CHOFFS      * ILLEGAL CHAR.
     4656 6000
0597 4658 10BA          JMP  DECOD8
0598 465A 0287     DECOD4 CI   R7,ERRORC      RECV.DATA= ERROR CODE ?
     465C 8000
0599 465E 16F8          JNE  DECODX
0600 4660 0202     DECSP1 LI   R2,10          R2= ERROR MESSAGE INDEX
     4662 000A
0601 4664 1005          JMP  DECSP2
0602               *DECOD6 CI   R7,BREAKC      RECV.DATA= BREAK CODE ?
0603               *    JNE  DECODX
0604               *    LI   R2,8           R2= BREAK MESSAGE INDEX
0605               *    JMP  DECSP2
0606               *
0607 4666 0972     DECSP  SRL  R2,7           R2= 0, 2, 4, 6 ..
0608 4668 13F3          JEQ  DECODX         R2 CANNOT BE 0
0609 466A 0282          CI   R2,10          R2 CANNOT BE > 10
     466C 000A
0610 466E 15F0          JGT  DECODX
0611 4670 C2A2     DECSP2 MOV  @SPMES(R2),R10  GET START OF MESSAGE
     4672 4BDA'
0612 4674 0286          CI   R6,RR6MIN      CURSOR= BEGINNING OF LINE
     4676 039A
0613 4678 1306          JEQ  DECSP3         GO IF YES
```

```
0614 467A 0200          LI    R0,RECLIN-1   R0= # OF LINES TO SCROLL
     467C 0009
0615 467E 0201          LI    R1,RSCRNO     R1= TOP OF RECV.SCREEN
     4680 0232
0616 4682 06A0          BL    @SCROLL       GO SCROLL RECV.SCREEN
     4684 4754'
0617 4686 C046   DECSP3 MOV   R6,R1         ACCESS CURSOR
0618 4688 0221          AI    R1,WDBIT
     468A 4000
0619 468C 06A0          BL    @VDPWAD
     468E 4704'
0620 4690 C07A          MOV   *R10+,R1      R1= # OF CHAR. OF THE MESSAGE
0621 4692 A181          A     R1,R6         CURSOR= END OF MESSAGE + 1
0622 4694 0586          INC   R6
0623 4696 D0FA   DECSP4 MOVB  *R10+,R3      GET 1 CHAR.OF THE MESSAGE
0624 4698 0223          AI    R3,CHOFFS     ADD OFFSET TO P.G.T.
     469A 6000
0625 469C DBC3          MOVB  R3,@VWD(R15)  DISPLAY 1 CHAR.
     469E FFFE
0626 46A0 0601          DEC   R1            LOOP TILL ALL DONE
0627 46A2 16F9          JNE   DECSP4
0628 46A4 022A          AI    R10,-6        INDEX - 6
     46A6 FFFA
0629 46A8 028A          CI    R10,SKMES     MESSAGE IS END OF XMIT ?
     46AA 4BFC'
0630 46AC 1303          JEQ   DECSP5
0631 46AE 028A          CI    R10,ARMES     MESSAGE IS END OF MESSAGE ?
     46B0 4BF6'
0632 46B2 16A4          JNE   DECOD7
0633                    *
0634 46B4 0200   DECSP5 LI    R0,RECLIN-1   R0= # OF LINES TO SCROLL
     46B6 0009
0635 46B8 0201          LI    R1,RSCRNO     R1= TOP OF RECV.SCREEN
     46BA 0232
0636 46BC 06A0          BL    @SCROLL       GO SCROLL RECV.SCREEN
     46BE 4754'
0637 46C0 C064          MOV   @CHACNT(R4),R1  R1= CHAR.COUNT
     46C2 FF36
0638 46C4 3860          MPY   @D12,R1       R1 & R2 = CHAR.COUNT * 12
     46C6 4D30'
0639 46C8 C024          MOV   @SEC(R4),R0   R0= SECONDS ELAPSED
     46CA FF30
0640 46CC C0C0          MOV   R0,R3
0641 46CE 0913          SRL   R3,1          ADD DIVISOR/2 TO DIVIDENT
0642 46D0 A083          A     R3,R2         * TO AVOID ROUND OFF
0643 46D2 3C40          DIV   R0,R1         R1 & R2 / R0
0644 46D4 04C0          CLR   R0            R0 & R1 = 0 & WORDS/MIN
0645 46D6 3C20          DIV   @D10,R0       R0 & R1 = TENS & UNITS DIGIT
     46D8 4D32'
0646 46DA C081          MOV   R1,R2         TEMP.STORE UNITS DIGIT
0647 46DC 0201          LI    R1,RWPM+WDBIT  ACCESS RECV.WORDS/MIN ADDR.
     46DE 41F9
0648 46E0 06A0          BL    @VDPWAD
     46E2 4704'
0649 46E4 06C0          SWPB  R0            DISPLAY TENS DIGIT
0650 46E6 0220          AI    R0,CHOFFS
     46E8 6000
0651 46EA 0220          AI    R0,'0'*256
     46EC 3000
0652 46EE DBC0          MOVB  R0,@VWD(R15)
```

```
      46F0 FFFE
 0653 46F2 06C2          SWPB R2              DISPLAY UNITS DIGIT
 0654 46F4 0222          AI   R2,CHOFFS
      46F6 6000
 0655 46F8 0222          AI   R2,'0'*256
      46FA 3000
 0656 46FC DBC2          MOVB R2,@VWD(R15)
      46FE FFFE
 0657 4700 0460          B    @RECV           TO BEGINNING OF RECEIVER
      4702 43A2'
```

```
0659                *
0660                *      *** SEND ADDRESS STORED IN R1 TO VDP CHIP
0661                *
0662 4704 06C1  VDPWAD SWPB R1
0663 4706 D7C1         MOVB R1,*R15       WRITE LOWER BYTE ADDR.
0664 4708 06C1         SWPB R1
0665 470A D7C1         MOVB R1,*R15       WRITE HIGHER BYTE ADDR.
0666 470C 045B         RT
```

```
                 *
                 *       *** DISPLAY A SERIES OF MESSAGES ON THE SCREEN.
                 *       ***      R5 POINTS TO THE FIRST MESSAGE.
                 *
 470E C035  MESDSP MOV   *R5+,R0       R0= NEXT MESSAGE LINK
 4710 C075         MOV   *R5+,R1       R1= ADDRESS OF 1ST CHAR.ON SCREEN
 4712 D0B5         MOVB  *R5+,R2       R2= # OF CHAR.  IN THIS MESSAGE
 4714 0982         SRL   R2,8          PUT TO LOWER BYTE
 4716 0221         AI    R1,WDBIT      ADD CONTROL BIT
 4718 4000
 471A 06C1         SWPB  R1
 471C D7C1         MOVB  R1,*R15       WRITE LOWER BYTE ADDRESS
 471E 06C1         SWPB  R1
 4720 D7C1         MOVB  R1,*R15       WRITE HIGHER BYTE ADDRESS
 4722 D0F5  MESDS1 MOVB  *R5+,R3       GET ASCII CHAR. IN TEXT
 4724 0223         AI    R3,CHOFFS     ADD OFFSET TO P.G.T.
 4726 6000
 4728 DBC3         MOVB  R3,@VWD(R15)  DISPLAY 1 CHAR.
 472A FFFE
 472C 0602         DEC   R2            THIS MESSAGE DONE ?
 472E 16F9         JNE   MESDS1
 4730 C140         MOV   R0,R5         R5= LINK TO NEXT MESSAGE
 4732 16ED         JNE   MESDSP        EXIT IF R5= 0
 4734 045B         RT
```

```
0690                *
0691                *       *** THIS SUBROUTINE BLANKS THE ENTIRE SCREEN
0692                *
0693 4736 0201   BLANKS LI    R1,0+WDBIT     R1= TOP OF SCREEN + CONTROL BIT
     4738 4000
0694 473A 06C1          SWPB R1
0695 473C D7C1          MOVB R1,*R15      WRITE LOWER BYTE ADDR.
0696 473E 06C1          SWPB R1
0697 4740 D7C1          MOVB R1,*R15      WRITE HIGHER BYTE ADDR.
0698 4742 D160          MOVB @BLANKC,R5   R5= BLANK CODE
     4744 4D18'
0699 4746 0201          LI    R1,SCNMAX    R1= # OF CHAR.TO BLANK
     4748 03C0
0700 474A DBC5   BLANK1 MOVB R5,@VWD(R15) BLANK 1 CHAR.ON SCREEN
     474C FFFE
0701 474E 0601          DEC   R1           LOOP TILL ALL DONE
0702 4750 16FC          JNE   BLANK1
0703 4752 045B          RT
```

```
0705                *
0706                *       ***    SCROLL XMIT/RECV HALF SCREEN
0707                *
0708                *       *** R0= # OF LINES TO SCROLL, R1= TOP OF SCREEN,
0709                *       *** R6= CURRENT CURSOR.
0710                *
0711 4754 0202      SCROLL LI   R2,40
     4756 0028
0712 4758 A081             A    R1,R2          R2= 1ST CHAR.OF SOURCE LINE
0713 475A 0221             AI   R1,WDBIT       ADD CONTROL BIT
     475C 4000
0714 475E 06C2      SCROL6 SWPB R2             ACCESS 1ST CHAR.OF SOURCE LINE
0715 4760 D7C2             MOVB R2,*R15
0716 4762 06C2             SWPB R2
0717 4764 D7C2             MOVB R2,*R15
0718 4766 0203             LI   R3,RAMBUF
     4768 FF40
0719 476A A0C4             A    R4,R3          R3= CPU RAM TEMP.BUFFER
0720 476C 0207             LI   R7,COLMAX      R7= # OF CHAR.TO MOVE AT ONCE
     476E 0024
0721 47/0 DCEF      SCROL1 MOVB @VRD(R15),*R3+   MOVE 36 CHAR.FROM VDP RAM
     4772 FBFE
0722 4774 0607             DEC  R7                  * TO CPU RAM
0723 47/6 16FC             JNE  SCROL1
0724 4778 06C1             SWPB R1             ACCESS 1ST CHAR.OF DESTINATION
0725 47/A D7C1             MOVB R1,*R15        * LINE
0726 47/C 06C1             SWPB R1
0727 47/E D7C1             MOVB R1,*R15
0728 4780 0203             LI   R3,RAMBUF
     4782 FF40
0729 4784 A0C4             A    R4,R3          R3= CPU RAM TEMP.BUFFER
0730 4786 0207             LI   R7,COLMAX
     4788 0024
0731 478A DBF3      SCROL2 MOVB *R3+,@VWD(R15)   MOVE 36 CHAR.FROM CPU RAM
     478C FFFE
0732 478E 0607             DEC  R7                  * TO VDP RAM
0733 4790 16FC             JNE  SCROL2
0734 4792 D1E4             MOVB @FLGREG(R4),R7   GET FLAG REG.
     4794 FF26
0735 4796 21E0             COC  @FORCE,R7      FORCED FLAG SET  ?
     4798 4D48'
0736 479A 131C             JEQ  SCROL3         GO IF YES
0737 479C C1CB      SCROL4 MOV  R11,R7         SAVE RETURN ADDRESS
0738 479E 06A0             BL   @HZ250         CHECK OSCILLATOR IN
     47A0 4814'
0739 47A2 C2C7             MOV  R7,R11         RESTORE RETURN ADDRESS
0740 47A4 0221             AI   R1,40          NEXT DESTINATION LINE
     47A6 0028
0741 47A8 0222             AI   R2,40          NEXT SOURCE LINE
     47AA 0028
0742 47AC 0600             DEC  R0             ALL LINES DONE ?
0743 47AE 16D7             JNE  SCROL6
0744 47B0 0222             AI   R2,-40+WDBIT BACK 1 LINE FOR WRITE-DATA
     47B2 3FD8
0745 47B4 06C2             SWPB R2             ACCESS 1ST CHAR.OF LAST LINE
0746 47B6 D7C2             MOVB R2,*R15
0747 47B8 06C2             SWPB R2
0748 47BA D7C2             MOVB R2,*R15
0749 47BC 0207             LI   R7,COLMAX
     47BE 0024
```

```
0750 47C0 D0E0          MOVB @BLANKC,R3    R3= BLANK CODE
     47C2 4D18'
0751 47C4 DBC3   SCROL8 MOVB R3,@VWD(R15) BLANK THE LAST LINE
     47C6 FFFE
0752 47C8 0607          DEC  R7
0753 47CA 16FC          JNE  SCROL8
0754 47CC 0222          AI   R2,-WDBIT     GET RID OF CONTROL BIT
     47CE C000
0755 47D0 C182          MOV  R2,R6         R6= NEW CURSOR VALUE
0756 47D2 045B          RT
0757 47D4 1F00   SCROL3 TB   0             KEYER DOWN  ?
0758 47D6 16E2          JNE  SCROL4        GO IF NOT
0759 47D8 04CC          CLR  R12           ACCESS 9901 TIMER
0760 47DA 33E0          LDCR @MAXTIM,15    RESET TIMER TO MAX
     47DC 4D40'
0761 47DE 1E00          SBZ  0             FREE CLOCK BUFFER FOR UPDATE
0762 47E0 04E4          CLR  @ACCUM(R4)    CLR 250 HZ ACCUM.
     47E2 FF32
0763 47E4 04E4          CLR  @SUM901(R4)   CLR SUM OF 9901 TIMER DATA
     47E6 FF70
0764 47E8 D820          MOVB @ENTONE,@SGCADR  ENABLE SOUND CHIP
     47EA 4D3C'
     47EC 8400
0765 47EE 5920          SZCB @FORCE,@FLGREG(R4)  CLR FORCED FLAG
     47F0 4D48'
     47F2 FF26
0766 47F4 F920          SOCB @SEMIFR,@FLGREG(R4) SET SEMI-FORCED FLAG
     47F6 4D44'
     47F8 FF26
0767 47FA C324          MOV  @SAVR12(R4),R12  RESTORE CRU REG.
     47FC FF2C
0768 47FE 10CE          JMP  SCROL4
```

```
0770                *
0771                *      *** THIS SUBROUTINE CHECKS IF SHIFT-C IS
0772                *      *** PRESSED AND RETURNS PROPER STATUS
0773                *
0774 4800 020C SHC  LI   R12,>24        ACCESS KEYBOARD
     4802 0024
0775 4804 30E0      LDCR @H0000,3       TEST COLUMN 1
     4806 4D26'
0776 4808 1FF5      TB   -11            SHIFT KEY PRESSED ?
0777 480A 1303      JEQ  SHCX           EXIT IF NOT
0778 480C 30E0      LDCR @H03,3         TEST COLUMN 4
     480E 4D1E'
0779 4810 1FF5      TB   -11            C KEY PRESSED ?
0780 4812 045B SHCX RT
```

```
0782                  *
0783                  *       *** TEST 250 HERTZ LINE TO UPDATE TIMING
0784                  *       *** WHEN A RISING EDGE TRANSITION IS SENSED.
0785                  *       *** R8 IS USED TO TEMP.STORE INPUT AND FLGREG.
0786                  *
0787 4814 1F01  HZ250   TB   1               TEST INPUT LINE
0788 4816 1302          JEQ  HZ250A
0789 4818 04C8          CLR  R8              R8= LOW INDICATION
0790 481A 1001          JMP  HZ250B
0791 481C 0708  HZ250A  SETO R8              R8= HIGH INDICATION
0792 481E D224  HZ250B  MOVB @FLGREG(R4),R8  HIGH BYTE(R8)= FLAG REG.
     4820 FF26
0793 4822 2220          COC  @RTFLAG,R8      REALTIME FLAG SET ?
     4824 4D46'
0794 4826 1612          JNE  HZ250C          EXIT IF NOT
0795 4828 06C8          SWPB R8              GET INPUT IN LOW BYTE
0796 482A D208          MOVB R8,R8           NEW INPUT= HIGH ?
0797 482C 1310          JEQ  HZ250X          GO IF NOT
0798 482E 9920          CB   @H0000,@OLDIN(R4)  OLD INPUT= LOW ?
     4830 4D26'
     4832 FF38
0799 4834 160C          JNE  HZ250X          GO IF NOT
0800 4836 05A4          INC  @ACCUM(R4)      INCR 250 HZ ACCUM.
     4838 FF32
0801 483A 0624          DEC  @SUBSEC(R4)     SUBSECOND COUNT= 0 ?
     483C FF2E
0802 483E 1607          JNE  HZ250X          1 SEC.HASN'T ELAPSED IF NOT 0
0803 4840 C920          MOV  @D250,@SUBSEC(R4)  RESET SUB.SEC.COUNT
     4842 4D3A'
     4844 FF2E
0804 4846 05A4          INC  @SEC(R4)        INCR.SECOND COUNT
     4848 FF30
0805 484A 1001          JMP  HZ250X
0806 484C 06C8  HZ250C  SWPB R8              HIGH BYTE= INPUT
0807 484E D908  HZ250X  MOVB R8,@OLDIN(R4)  UPDATE OLD INPUT
     4850 FF38
0808 4852 045B          RT
```

```
0810                  *
0811                  *       *** UPON ENTRY OF THIS ROUTINE :
0812                  *       *** R8= TIME INTERVAL BASED ON 4 MILLI.SEC.
0813                  *       *** R9 & R10 = TIME INTERVAL BASED ON 22.3 MU.SEC.
0814                  *
0815                  *       *** IF TIME INTERVAL IS LESS THAN 300 MILLI.SEC.
0816                  *       *** R9 & R10 IS USED, OTHERWISE R8 IS USED.
0817                  *
0818                  *       *** THIS ROUTINE RETURNS :
0819                  *       *** R1= TIME INTERVAL BASED ON 1 MILLI.SEC.
0820                  *
0821 4854 C224  SCALE  MOV   @ACCUM(R4),R8   GET 250 HZ ACCUM.
     4856 FF32
0822 4858 0A28         SLA   R8,2         R8= TIME BASED ON 1 M.SEC.
0823 485A 0288         CI    R8,300       R8 < 300 M.SEC.  ?
     485C 012C
0824 485E 1102         JLT   SCALE1       GO IF YES
0825 4860 C048         MOV   R8,R1        USE 250 HZ ACCUM.
0826 4862 1008         JMP   SCALE2       EXIT
0827 4864 C2A4  SCALE1 MOV   @SUM901(R4),R10  R10= TOTAL 9901 TIMER DATA
     4866 FF70
0828 4868 04C9         CLR   R9           * CYCLES ELAPSED
0829 486A 022A         AI    R10,47/2     ADD DIVISOR/2 TO DIVIDENT
     486C 0017
0830 486E 3E60         DIV   @D47,R9      * TO AVOID ROUND OFF
     4870 4D2E'
0831 4872 C049         MOV   R9,R1        R1= DURATION BASED ON 1 M.SEC.
0832 4874 04E4  SCALE2 CLR   @SUM901(R4)  CLR SUM OF 9901 DATA
     4876 FF70
0833 4878 04E4         CLR   @ACCUM(R4)   CLR 250 HZ ACCUM
     487A FF32
0834 487C 045B         RT
```

```
0836                 *
0837                 *       *** DELAY 10 M.SEC. AND RETURN KEYER STATUS
0838                 *
0839 487E 020C  DEBOUN LI    R12,2          READ 9901 TIMER
     4880 0002
0840 4882 04CA         CLR   R10
0841 4884 1DFF         SBO   -1
0842 4886 378A         STCR  R10,14
0843 4888 1EFF         SBZ   -1
0844 488A 880A         C     R10,@MSEC10    HAS 10 M.SEC. ELAPSED ?
     488C 4D42'
0845 488E 1105         JLT   DEBOU1         GO IF YES
0846 4890 C28B         MOV   R11,R10        SAVE RETURN ADDR.
0847 4892 06A0         BL    @HZ250         CHECK OSCILLATOR IN
     4894 4814'
0848 4896 C2CA         MOV   R10,R11        RESTORE RETURN ADDR.
0849 4898 10F2         JMP   DEBOUN         CHECK 9901 TIMER
0850 489A C324  DEBOU1 MOV   @SAVR12(R4),R12  RESTORE CRU REG.
     489C FF2C
0851 489E 1F00         TB    0              CHECK KEYER STATUS
0852 48A0 045B         RT                   RETURN TO CALLER
```

```
0854                *
0855                *       *** CHECK KEYBOADR TO ADJUST OFFSET
0856                *       *** BETWEEN CHAR. & WORD
0857                *
0858 48A2 020C  CHWRD  LI   R12,>24        KEY 'D' PRESSED ?
     48A4 0024
0859 48A6 30E0         LDCR @H03,3
     48A8 4D1E'
0860 48AA 1FF6         TB   -10
0861 48AC 130F         JEQ  CHWRD1         GO IF NOT
0862 48AE 0664         DECT @OFFSET(R4)    OFFSET - 2
     48B0 FF66
0863 48B2 1512         JGT  CHWRDX         GO IF > 0
0864 48B4 1311         JEQ  CHWRDX         GO IF = 0
0865 48B6 C224         MOV  @OFFSET(R4),R8   ABS(OFFSET) TOO LARGE ?
     48B8 FF66
0866 48BA 0748         ABS  R8
0867 48BC C1E4         MOV  @TWOEOC(R4),R7   UPPER LIMIT IS TWOEOC/4
     48BE FF6C
0868 48C0 0927         SRL  R7,2           R7= TWOEOC/4
0869 48C2 81C8         C    R8,R7          ABS(OFFSET) < TWOEOC/4 ?
0870 48C4 1109         JLT  CHWRDX         GO IF YES
0871 48C6 05E4         INCT @OFFSET(R4)    PUT OFFSET BACK
     48C8 FF66
0872 48CA 045B         RT
0873 48CC 30E0  CHWRD1 LDCR @H02,3         KEY 'S' PRESSED ?
     48CE 4D20'
0874 48D0 1FF6         TB   -10
0875 48D2 1302         JEQ  CHWRDX         GO IF NOT
0876 48D4 05E4         INCT @OFFSET(R4)    OFFSET + 2
     48D6 FF66
0877 48D8 045B  CHWRDX RT
```

```
0879                *
0880                *       *** BEFORE SOFTWARE POWER-UP RESET, THIS DSR
0881                *       *** ROM MUST BE DISABLED. THE INSTRUCTIONS TO
0882                *       *** DISABLE THE ROM ARE LOADED INTO CPU RAM
0883                *       *** FOR EXECUTION
0884                *
0885 48DA 0200     RESET0 LI    R0,RETURN     R0= INSTR. TO BE LOADED
     48DC 48F4'
0886 48DE 0202            LI    R2,5          5 WORDS TO LOAD
     48E0 0005
0887 48E2 0201            LI    R1,STOCUR
     48E4 FF20
0888 48E6 0221            AI    R1,PAD        R1= START OF CPU RAM
     48E8 83E0
0889 48EA C0C1            MOV   R1,R3         TEMP. STORAGE
0890 48EC CC70     RESET1 MOV   *R0+,*R1+     MOV ROM CODES TO CPU RAM
0891 48EE 0602            DEC   R2            REPEAT 5 TIMES
0892 48F0 16FD            JNE   RESET1
0893 48F2 0453            B     *R3           START EXECUTING CODES IN RAM
0894                *
0895 48F4 C324     RETURN MOV   @SAVR12(R4),R12  RESTORE CRU REG.
     48F6 FF2C
0896 48F8 1E00            SBZ   0                DISABLE DSR ROM
0897 48FA 0460            B     @RSTADR       LOAD POWER-UP WORK SPACE..
     48FC 0012
```

```
0899                 *       *** MESSAGE FOR TI HEADER  ***
0900                 *
0901 48FE 4914' DISPTI DATA DSPTI1,0+11  NEXT MESSAGE, SCREEN LOCATION
     4900 000B
0902 4902   11              BYTE 17            # OF CHAR. TO DISPLAY
0903 4903   54              TEXT 'TEXAS INSTRUMENTS'
     4904   45
     4905   58
     4906   41
     4907   53
     4908   20
     4909   49
     490A   4E
     490B   53
     490C   54
     490D   52
     490E   55
     490F   4D
     4910   45
     4911   4E
     4912   54
     4913   53
0904 4914                   EVEN
0905 4914 4926' DSPTI1 DATA DSPTI2,40+13 NEXT MESSAGE, SCREEN LOCATION
     4916 0035
0906 4918   0D              BYTE 13            # OF CHAR. TO DISPLAY
0907 4919   48              TEXT 'HOME COMPUTER'
     491A   4F
     491B   4D
     491C   45
     491D   20
     491E   43
     491F   4F
     4920   4D
     4921   50
     4922   55
     4923   54
     4924   45
     4925   52
0908 4926                   EVEN
0909 4926 0000  DSPTI2 DATA 0,80+4         LAST MESSAGE, SCREEN LOCATION
     4928 0054
0910 492A   1F              BYTE 31            # OF CHAR. TO DISPLAY
0911 492B   4D              TEXT 'MORSE CODE RECEIVER/TRANSMITTER'
     492C   4F
     492D   52
     492E   53
     492F   45
     4930   20
     4931   43
     4932   4F
     4933   44
     4934   45
     4935   20
     4936   52
     4937   45
     4938   43
     4939   45
     493A   49
     493B   56
     493C   45
     493D   52
     493E   2F
     493F   54
     4940   52
     4941   41
     4942   4E
     4943   53
     4944   4D
     4945   49
     4946   54
     4947   54
     4948   45
     4949   52
0912 494A                 EVEN
```

```
0914                *
0915                *          *** MESSAGES FOR 1ST MENU SELECTION
0916                *
0917 494A 4960' MENU1    DATA MENU11,160+4 NEXT MESSAGE, SCREEN LOCATION
     494C 00A4
0918 494E   10          BYTE 16            # OF CHAR. TO DISPLAY
0919 494F   54          TEXT 'TONE SELECTION :'
     4950   4F
     4951   4E
     4952   45
     4953   20
     4954   53
     4955   45
     4956   4C
     4957   45
     4958   43
     4959   54
     495A   49
     495B   4F
     495C   4E
     495D   20
     495E   3A
0920 4960               EVEN
0921 4960 496A' MENU11  DATA MENU12,200+4 NEXT MESSAGE, SCREEN LOCATION
     4962 00CC
0922 4964   05          BYTE 5             # OF CHAR. TO DISPLAY
0923 4965   50          TEXT 'PRESS'
     4966   52
     4967   45
     4968   53
     4969   53
0924 496A               EVEN
0925 496A 4980' MENU12  DATA MENU13,240+4
     496C 00F4
0926 496E   10          BYTE 16
0927 496F   31          TEXT '1 FOR  300 HERTZ'
     4970   20
     4971   46
     4972   4F
     4973   52
     4974   20
     4975   20
     4976   33
     4977   30
     4978   30
     4979   20
     497A   48
     497B   45
     497C   52
     497D   54
     497E   5A
0928 4980               EVEN
0929 4980 4996' MENU13  DATA MENU14,280+4
     4982 011C
0930 4984   10          BYTE 16
0931 4985   32          TEXT '2 FOR  600 HERTZ'
     4986   20
     4987   46
     4988   4F
     4989   52
     498A   20
     498B   20
     498C   36
     498D   30
     498E   30
     498F   20
     4990   48
     4991   45
     4992   52
     4993   54
     4994   5A
0932 4996               EVEN
0933 4996 49AC' MENU14 DATA MENU15,320+4
     4998 0144
0934 499A   10          BYTE 16
0935 499B   33          TEXT '3 FOR 1200 HERTZ'
     499C   20
     499D   46
     499E   4F
     499F   52
     49A0   20
     49A1   31
     49A2   32
     49A3   30
     49A4   30
     49A5   20
     49A6   48
     49A7   45
     49A8   52
     49A9   54
     49AA   5A
0936 49AC               EVEN
0937 49AC 0000  MENU15 DATA 0,360+4      LAST MESSAGE, SCREEN LOCATION
     49AE 016C
0938 49B0   10          BYTE 16
0939 49B1   34          TEXT '4 FOR 2400 HERTZ'
     49B2   20
     49B3   46
     49B4   4F
     49B5   52
     49B6   20
     49B7   32
     49B8   34
     49B9   30
     49BA   30
     49BB   20
     49BC   48
     49BD   45
     49BE   52
     49BF   54
     49C0   5A
0940 49C2               EVEN
```

```
0942                *
0943                *       *** MESSAGES FOR 2ND MENU SELECTION
0944                *
0945 49C2 49E4' MENU2   DATA MENU21,160+4
     49C4 00A4
0946 49C6   1D          BYTE 29
0947 49C7   54          TEXT 'TRANSMITTER SPEED SELECTION :'
     49C8   52
     49C9   41
     49CA   4E
     49CB   53
     49CC   4D
     49CD   49
     49CE   54
     49CF   54
     49D0   45
     49D1   52
     49D2   20
     49D3   53
     49D4   50
     49D5   45
     49D6   45
     49D7   44
     49D8   20
     49D9   53
     49DA   45
     49DB   4C
     49DC   45
     49DD   43
     49DE   54
     49DF   49
     49E0   4F
     49E1   4E
     49E2   20
     49E3   3A
0948 49E4               EVEN
0949 49E4 49EE' MENU21  DATA MENU22,200+4
     49E6 00CC
0950 49E8   05          BYTE 5
0951 49E9   50          TEXT 'PRESS'
     49EA   52
     49EB   45
     49EC   53
     49ED   53
0952 49EE               EVEN
0953 49EE 4A08' MENU22  DATA MENU23,240+4
     49F0 00F4
0954 49F2   15          BYTE 21
0955 49F3   31          TEXT '1 FOR  5 WORDS/MINUTE'
     49F4   20
     49F5   46
     49F6   4F
     49F7   52
     49F8   20
     49F9   20
     49FA   35
     49FB   20
     49FC   57
     49FD   4F
     49FE   52
     49FF   44
     4A00   53
     4A01   2F
     4A02   4D
     4A03   49
     4A04   4E
     4A05   55
     4A06   54
     4A07   45
0956 4A08                 EVEN
0957 4A08 4A22' MENU23 DATA MENU24,280+4
     4A0A 011C
0958 4A0C   15            BYTE 21
0959 4A0D   32            TEXT '2 FOR 10 WORDS/MINUTE'
     4A0E   20
     4A0F   46
     4A10   4F
     4A11   52
     4A12   20
     4A13   31
     4A14   30
     4A15   20
     4A16   57
     4A17   4F
     4A18   52
     4A19   44
     4A1A   53
     4A1B   2F
     4A1C   4D
     4A1D   49
     4A1E   4E
     4A1F   55
     4A20   54
     4A21   45
0960 4A22                 EVEN
0961 4A22 4A3C' MENU24 DATA MENU25,320+4
     4A24 0144
0962 4A26   15            BYTE 21
0963 4A27   33            TEXT '3 FOR 12 WORDS/MINUTE'
     4A28   20
     4A29   46
     4A2A   4F
     4A2B   52
     4A2C   20
     4A2D   31
     4A2E   32
     4A2F   20
     4A30   57
     4A31   4F
     4A32   52
     4A33   44
     4A34   53
     4A35   2F
     4A36   4D
     4A37   49
     4A38   4E
     4A39   55
     4A3A   54
     4A3B   45
0964 4A3C                 EVEN
```

```
0965 4A3C 4A56' MENU25 DATA MENU26,360+4
     4A3E 016C
0966 4A40   15         BYTE 21
0967 4A41   34         TEXT '4 FOR 15 WORDS/MINUTE'
     4A42   20
     4A43   46
     4A44   4F
     4A45   52
     4A46   20
     4A47   31
     4A48   35
     4A49   20
     4A4A   57
     4A4B   4F
     4A4C   52
     4A4D   44
     4A4E   53
     4A4F   2F
     4A50   4D
     4A51   49
     4A52   4E
     4A53   55
     4A54   54
     4A55   45
0968 4A56              EVEN
0969 4A56 4A70' MENU26 DATA MENU27,400+4
     4A58 0194
0970 4A5A   15         BYTE 21
0971 4A5B   35         TEXT '5 FOR 20 WORDS/MINUTE'
     4A5C   20
     4A5D   46
     4A5E   4F
     4A5F   52
     4A60   20
     4A61   32
     4A62   30
     4A63   20
     4A64   57
     4A65   4F
     4A66   52
     4A67   44
     4A68   53
     4A69   2F
     4A6A   4D
     4A6B   49
     4A6C   4E
     4A6D   55
     4A6E   54
     4A6F   45
0972 4A70              EVEN
0973 4A70 4A8A' MENU27 DATA MENU28,440+4
     4A72 01BC
0974 4A74   15         BYTE 21
0975 4A75   36         TEXT '6 FOR 25 WORDS/MINUTE'
     4A76   20
     4A77   46
     4A78   4F
     4A79   52
     4A7A   20
     4A7B   32
     4A7C   35
     4A7D   20
     4A7E   57
     4A7F   4F
     4A80   52
     4A81   44
     4A82   53
     4A83   2F
     4A84   4D
     4A85   49
     4A86   4E
     4A87   55
     4A88   54
     4A89   45
0976 4A8A                  EVEN
0977 4A8A 0000   MENU28 DATA 0,480+4
     4A8C 01E4
0978 4A8E   15             BYTE 21
0979 4A8F   37             TEXT '7 FOR 30 WORDS/MINUTE'
     4A90   20
     4A91   46
     4A92   4F
     4A93   52
     4A94   20
     4A95   33
     4A96   30
     4A97   20
     4A98   57
     4A99   4F
     4A9A   52
     4A9B   44
     4A9C   53
     4A9D   2F
     4A9E   4D
     4A9F   49
     4AA0   4E
     4AA1   55
     4AA2   54
     4AA3   45
0980 4AA4                  EVEN
```

```
0982                 *
0983                 *       *** MESSAGES OF 3RD MENU SELECTION
0984                 *
0985 4AA4 4ABC' MENU3   DATA MENU31,160+4
     4AA6 00A4
0986 4AA8   13          BYTE 19
0987 4AA9   43          TEXT 'CONTROL SELECTION :'
     4AAA   4F
     4AAB   4E
     4AAC   54
     4AAD   52
     4AAE   4F
     4AAF   4C
     4AB0   20
     4AB1   53
     4AB2   45
     4AB3   4C
     4AB4   45
     4AB5   43
     4AB6   54
     4AB7   49
     4AB8   4F
     4AB9   4E
     4ABA   20
     4ABB   3A
0988 4ABC               EVEN
0989 4ABC 4AC6' MENU31  DATA MENU32,200+4
     4ABE 00CC
0990 4AC0   05          BYTE 5
0991 4AC1   50          TEXT 'PRESS'
     4AC2   52
     4AC3   45
     4AC4   53
     4AC5   53
0992 4AC6               EVEN
0993 4AC6 4AE0' MENU32  DATA MENU33,240+4
     4AC8 00F4
0994 4ACA   14          BYTE 20
0995 4ACB   53          TEXT 'SHIFT-R FOR RECEIVER'
     4ACC   48
     4ACD   49
     4ACE   46
     4ACF   54
     4AD0   2D
     4AD1   52
     4AD2   20
     4AD3   46
     4AD4   4F
     4AD5   52
     4AD6   20
     4AD7   52
     4AD8   45
     4AD9   43
     4ADA   45
     4ADB   49
     4ADC   56
     4ADD   45
     4ADE   52
0996 4AE0               EVEN
0997 4AE0 4AFC' MENU33  DATA MENU34,280+4
```

```
     4AF2 011C
0998 4AE4   17         BYTE 23
0999 4AF5   53         TEXT 'SHIFT-T FOR TRANSMITTER'
     4AE6   48
     4AE7   49
     4AE8   46
     4AF9   54
     4AFA   2D
     4AEB   54
     4AFC   20
     4AFD   46
     4AFE   4F
     4AEF   52
     4AF0   20
     4AF1   54
     4AF2   52
     4AF3   41
     4AF4   4E
     4AF5   53
     4AF6   4D
     4AF7   49
     4AF8   54
     4AF9   54
     4AFA   45
     4AFB   52
1000 4AFC              EVEN
1001 4AFC 4B1C' MENU34 DATA MENU35,320+4
     4AFE 0144
1002 4B00   1A         BYTE 26
1003 4B01   53         TEXT 'SHIFT-M FOR MENU SELECTION'
     4B02   48
     4B03   49
     4B04   46
     4B05   54
     4B06   2D
     4B07   4D
     4B08   20
     4B09   46
     4B0A   4F
     4B0B   52
     4B0C   20
     4B0D   4D
     4B0E   45
     4B0F   4E
     4B10   55
     4B11   20
     4B12   53
     4B13   45
     4B14   4C
     4B15   45
     4B16   43
     4B17   54
     4B18   49
     4B19   4F
     4B1A   4E
1004 4B1C              EVEN
1005 4B1C 4B3E' MENU35 DATA MENU36,360+4
     4B1E 016C
1006 4B20   1D         BYTE 29
1007 4B21   53         TEXT 'SHIFT-C FOR CONTROL SELECTION'
```

```
     4B22   48
     4B23   49
     4B24   46
     4B25   54
     4B26   2D
     4B27   43
     4B28   20
     4B29   46
     4B2A   4F
     4B2B   52
     4B2C   20
     4B2D   43
     4B2E   4F
     4B2F   4E
     4B30   54
     4B31   52
     4B32   4F
     4B33   4C
     4B34   20
     4B35   53
     4B36   45
     4B37   4C
     4B38   45
     4B39   43
     4B3A   54
     4B3B   49
     4B3C   4F
     4B3D   4E
1008 4B3E             EVEN
1009 4B3E 0000 MENU36 DATA 0,400+4
     4B40 0194
1010 4B42   18        BYTE 24
1011 4B43   53        TEXT 'SHIFT-Q FOR MASTER RESET'
     4B44   48
     4B45   49
     4B46   46
     4B47   54
     4B48   2D
     4B49   51
     4B4A   20
     4B4B   46
     4B4C   4F
     4B4D   52
     4B4E   20
     4B4F   4D
     4B50   41
     4B51   53
     4B52   54
     4B53   45
     4B54   52
     4B55   20
     4B56   52
     4B57   45
     4B58   53
     4B59   45
     4B5A   54
1012 4B5C             EVEN
```

```
1014                *
1015                *      *** SCREEN SEPARATOR
1016                *
1017 4B5C 4B84' XMTREC DATA XMTRE1,0+2
     4B5E 0002
1018 4B60   23         BYTE 35
1019 4B61   4D         TEXT 'MORSE CODE TRANSMITTER    WORDS/MIN'
     4B62   4F
     4B63   52
     4B64   53
     4B65   45
     4B66   20
     4B67   43
     4B68   4F
     4B69   44
     4B6A   45
     4B6B   20
     4B6C   54
     4B6D   52
     4B6E   41
     4B6F   4E
     4B70   53
     4B71   4D
     4B72   49
     4B73   54
     4B74   54
     4B75   45
     4B76   52
     4B77   20
     4B78   20
     4B79   20
     4B7A   20
     4B7B   57
     4B7C   4F
     4B7D   52
     4B7E   44
     4B7F   53
     4B80   2F
     4B81   4D
     4B82   49
     4B83   4E
1020 4B84              EVEN
1021 4B84 4BB2' XMTRE1 DATA XMTRE2,440
     4B86 01B8
1022 4B88   28         BYTE 40
1023 4B89   2A         TEXT '****************************************'
     4B8A   2A
     4B8B   2A
     4B8C   2A
     4B8D   2A
     4B8E   2A
     4B8F   2A
     4B90   2A
     4B91   2A
     4B92   2A
     4B93   2A
     4B94   2A
     4B95   2A
     4B96   2A
     4B97   2A
     4B98   2A
     4B99   2A
     4B9A   2A
     4B9B   2A
     4B9C   2A
     4B9D   2A
     4B9E   2A
     4B9F   2A
     4BA0   2A
     4BA1   2A
     4BA2   2A
     4BA3   2A
     4BA4   2A
     4BA5   2A
     4BA6   2A
     4BA7   2A
     4BA8   2A
     4BA9   2A
     4BAA   2A
     4BAB   2A
     4BAC   2A
     4BAD   2A
     4BAE   2A
     4BAF   2A
     4BB0   2A
1024 4BB2                EVEN
1025 4BB2 0000  XMTRE2 DATA 0,480+2
     4BB4 01E2
1026 4BB6   23           BYTE 35
1027 4BB7   4D           TEXT 'MORSE CODE RECEIVER       WORDS/MIN'
     4BB8   4F
     4BB9   52
     4BBA   53
     4BBB   45
     4BBC   20
     4BBD   43
     4BBE   4F
     4BBF   44
     4BC0   45
     4BC1   20
     4BC2   52
     4BC3   45
     4BC4   43
     4BC5   45
     4BC6   49
     4BC7   56
     4BC8   45
     4BC9   52
     4BCA   20
     4BCB   20
     4BCC   20
     4BCD   20
     4BCE   20
     4BCF   20
     4BD0   20
     4BD1   57
     4BD2   4F
     4BD3   52
     4BD4   44
     4BD5   53
     4BD6   2F
     4BD7   4D
     4BD8   49
     4BD9   4E
1028 4BDA              EVEN
```

```
1030                *
1031                *       *** MESSAGES WHEN SPECIAL CHAR. RECEIVED
1032                *
1033 4BDA 0000  SPMES   DATA >0000,ASMES   HEADER FIELD
     4BDC 4BF0'
1034 4BDE 4BF6'         DATA ARMES,SKMES   HEADER FIELD
     4BE0 4BFC'
1035 4BE2 4C02'         DATA BKMES,ERMES   HEADER FIELD
     4BE4 4BE6'
1036 4BE6 0007  ERMES   DATA 7             # OF CHAR. IN MESSAGE
1037 4BE8   21          TEXT '!ERROR!'     LAST WORD IS AN ERROR
     4BE9   45
     4BEA   52
     4BEB   52
     4BEC   4F
     4BED   52
     4BEE   21
1038 4BF0               EVEN
1039 4BF0 0004  ASMES   DATA 4
1040 4BF2   21          TEXT '!AS!'        WAIT
     4BF3   41
     4BF4   53
     4BF5   21
1041 4BF6               EVEN
1042 4BF6 0004  ARMES   DATA 4
1043 4BF8   21          TEXT '!AR!'        END OF MESSAGE
     4BF9   41
     4BFA   52
     4BFB   21
1044 4BFC               EVEN
1045 4BFC 0004  SKMES   DATA 4
1046 4BFE   21          TEXT '!SK!'        END OF TRANMISSION
     4BFF   53
     4C00   4B
     4C01   21
1047 4C02               EVEN
1048 4C02 0004  BKMES   DATA 4
1049 4C04   21          TEXT '!BK!'        BREAK
     4C05   42
     4C06   4B
     4C07   21
1050 4C08               EVEN
```

```
1052                 *
1053                 *       *** DATA FOR TONE 1 FREQUENCY
1054                 *
1055 4C08 8517  TONFRQ DATA >8517        300 HZ
1056 4C0A 8A0B         DATA >8A0B        600 HZ
1057 4C0C 8D05         DATA >8D05        1200 HZ
1058 4C0E 8E02         DATA >8E02        2400 HZ
1059                 *
1060                 *       *** DATA FOR XMIT SPEED SELECTION
1061                 *
1062                 *       *** EACH ENTRY CONTAINS 4 BYTES :
1063                 *       *** 1ST TWO BYTES ARE ASCII CODES OF WORDS/MIN.
1064                 *       *** 2ND TWO BYTES ARE COUNTDOWN FOR 9901 TIMER,
1065                 *       *** REPRESENTING DOT TIME.
1066                 *
1067 4C10 2035  XMTAB  DATA ' 5',>3FFF-8920    5 WORDS/MIN
     4C12 1D27
1068 4C14 3130         DATA '10',>3FFF-4460   10 WORDS/MIN
     4C16 2E93
1069 4C18 3132         DATA '12',>3FFF-3715   12 WORDS/MIN
     4C1A 317C
1070 4C1C 3135         DATA '15',>3FFF-2973   15 WORDS/MIN
     4C1E 3462
1071 4C20 3230         DATA '20',>3FFF-2230   20 WORDS/MIN
     4C22 3749
1072 4C24 3235         DATA '25',>3FFF-1784   25 WORDS/MIN
     4C26 3907
1073 4C28 3330         DATA '30',>3FFF-1653   30 WORDS/MIN
     4C2A 398A
```

```
          *
          *      *** DOT-DASH DATA FOR XMITTER
          *
          *      *** 1ST BYTE IS # OF EFFECTIVE BITS IN 2ND BYTE
          *      *** 2ND BYTE : 0= DOT, 1= DASH
          *
4C2C FF00 XMTDD  DATA >FF00        SPACE
4C2E 0000        DATA 0            !
4C30 0000        DATA 0            "
4C32 0000        DATA 0            #
4C34 0000        DATA 0            $
4C36 0000        DATA 0            %
4C38 0000        DATA 0            &
4C3A 0000        DATA 0            '
4C3C 06B4        DATA >06B4        (
4C3E 06B4        DATA >06B4        )
4C40 0000        DATA 0            *
4C42 0000        DATA 0            +
4C44 06CC        DATA >06CC        ,
4C46 0684        DATA >0684        -
4C48 0654        DATA >0654        .
4C4A 0000        DATA 0            /
4C4C 05F8        DATA >05F8        0
4C4E 0578        DATA >0578        1
4C50 0537        DATA >0537        2
4C52 0518        DATA >0518        3
4C54 0508        DATA >0508        4
4C56 0500        DATA >0500        5
4C58 0580        DATA >0580        6
4C5A 05C0        DATA >05C0        7
4C5C 05E0        DATA >05E0        8
4C5E 05F0        DATA >05F0        9
4C60 0000        DATA 0            :
4C62 0000        DATA 0            ;
4C64 0000        DATA 0            <
4C66 0000        DATA 0            =
4C68 0000        DATA 0            >
4C6A 0630        DATA >0630        ?
4C6C 0000        DATA 0            @
4C6E 0240        DATA >0240        A
4C70 0480        DATA >0480        B
4C72 04A0        DATA >04A0        C
4C74 0380        DATA >0380        D
4C76 0100        DATA >0100        E
4C78 0420        DATA >0420        F
4C7A 03C0        DATA >03C0        G
4C7C 0400        DATA >0400        H
4C7E 0200        DATA >0200        I
4C80 0470        DATA >0470        J
4C82 03A0        DATA >03A0        K
4C84 0440        DATA >0440        L
4C86 02C0        DATA >02C0        M
4C88 0280        DATA >0280        N
4C8A 03E0        DATA >03E0        O
4C8C 0460        DATA >0460        P
4C8E 04D0        DATA >04D0        Q
4C90 0340        DATA >0340        R
4C92 0300        DATA >0300        S
4C94 0180        DATA >0180        T
4C96 0320        DATA >0320        U
```

```
1135 4C98 0410          DATA >0410          V
1136 4C9A 0360          DATA >0360          W
1137 4C9C 0490          DATA >0490          X
1138 4C9E 04B0          DATA >04B0          Y
1139 4CA0 04C0          DATA >04C0          Z
1140 4CA2 0550   EOM    DATA >0550          END OF MESSAGE
```

```
1142                 *
1143                 *       *** RECEIVED DATA LOOK UP TABLE
1144                 *
1145 4CA4 0000  RECTAB DATA 0            0 : ADDRESS
1146 4CA6 4554         DATA 'ET'         2
1147 4CA8 4941         DATA 'IA'         4
1148 4CAA 4E4D         DATA 'NM'         6
1149 4CAC 5355         DATA 'SU'         8
1150 4CAE 5257         DATA 'RW'         A
1151 4CB0 444B         DATA 'DK'         C
1152 4CB2 474F         DATA 'GO'         E
1153 4CB4 4856         DATA 'HV'         10
1154 4CB6   46         BYTE 'F',0        12
     4CB7   00
1155 4CB8   4C         BYTE 'L',0        14
     4CB9   00
1156 4CBA 504A         DATA 'PJ'         16
1157 4CBC 4258         DATA 'BX'         18
1158 4CBE 4359         DATA 'CY'         1A
1159 4CC0 5A51         DATA 'ZQ'         1C
1160 4CC2 0000         DATA 0            1E
1161 4CC4 3534         DATA '54'         20
1162 4CC6   00         BYTE 0,'3'        22
     4CC7   33
1163 4CC8   00         BYTE 0,0          24
     4CC9   00
1164 4CCA   00         BYTE 0,'2'        26
     4CCB   32
1165 4CCC   01         BYTE 1,0          28
     4CCD   00
1166 4CCE   02         BYTE 2,0          2A
     4CCF   00
1167 4CD0   00         BYTE 0,0          2C
     4CD1   00
1168 4CD2   00         BYTE 0,'1'        2E
     4CD3   31
1169 4CD4 362D         DATA '6-'         30
1170 4CD6   2F         BYTE '/',0        32
     4CD7   00
1171 4CD8 0000         DATA 0            34
1172 4CDA 0000         DATA 0            36
1173 4CDC   37         BYTE '7',0        38
     4CDD   00
1174 4CDE   00         BYTE 0,0          3A
     4CDF   00
1175 4CE0   38         BYTE '8',0        3C
     4CE1   00
1176 4CE2   39         BYTE '9','0'      3E
     4CE3   30
1177 4CE4   05         BYTE 5,0          40
     4CE5   00
1178 4CE6   00         BYTE 0,0          42
     4CE7   00
1179 4CE8   00         BYTE 0,3          44
     4CE9   03
1180 4CEA 0000         DATA 0            46
1181 4CEC 0000         DATA 0            48
1182 4CEE 0000         DATA 0            4A
1183 4CF0   3F         BYTE '?',0        4C
     4CF1   00
```

```
1184 4CF2 0000          DATA 0              4E
1185 4CF4 0000          DATA 0              50
1186 4CF6 0000          DATA 0              52
1187 4CF8   00          BYTE 0,'.'          54
     4CF9   2E
1188 4CFA 0000          DATA 0              56
1189 4CFC 0000          DATA 0              58
1190 4CFE 0000          DATA 0              5A
1191 4D00 0000          DATA 0              5C
1192 4D02 0000          DATA 0              5E
1193 4D04 0000          DATA 0              60
1194 4D06 0000          DATA 0              62
1195 4D08 0000          DATA 0              64
1196 4D0A 0000          DATA 0              66
1197 4D0C 0000          DATA 0              68
1198 4D0E 0000          DATA 0              6A
1199 4D10   00          BYTE 0,'('          6C
     4D11   28
1200 4D12 0000          DATA 0              6E
1201 4D14 0000          DATA 0              70
1202 4D16   00          BYTE 0,','          72
     4D17   2C
```

```
  1204 4D18 8000   BLANKC DATA >8000         BLANK CODE FOR DISPLAY
  1205 4D1A F000   HF0    DATA >F000
  1206 4D1C 8100   H81    DATA >8100
  1207 4D1E 0300   H03    DATA >0300
  1208 4D20 0200   H02    DATA >0200
  1209 4D22 8700   H87    DATA >8700
  1210 4D24 CE00   HCE    DATA >CE00
  1211 4D26 0000   H0000  DATA >0000
  1212 4D28 7E00   CURSOR DATA >7E00         CURSOR CHAR.
  1213 4D2A 8000   H80    DATA >8000
  1214 4D2C 0028   D40    DATA 40
  1215 4D2E 002F   D47    DATA 47
  1216 4D30 000C   D12    DATA 12
  1217 4D32 000A   D10    DATA 10
  1218 4D34 0005   D5     DATA 5
  1219 4D36 0384   D900   DATA 900
  1220 4D38 012C   D300   DATA 300
  1221 4D3A 00FA   D250   DATA 250
  1222 4D3C 9000   ENTONE DATA >9000
  1223 4D3E 9F00   DISTON DATA >9F00
  1224 4D40 7FFF   MAXTIM DATA >7FFF
  1225 4D42 3E3F   MSEC10 DATA >3E3F         10 M.SEC.9901 CONTENT
  1226 4D44 0800   SEMIFR DATA >0800
  1227 4D46 1000   RTFLAG DATA >1000
  1228 4D48 2000   FORCE  DATA >2000
  1229 4D4A 4000   WORD   DATA >4000
  1230                    END
 NO ERRORS,     NO WARNINGS
```

```
LABEL     VALUE    DEFN   REFERENCES

ACCUM     FF32     0046   0418   0480   0762   0800   0821   0833
ARMES     4BF6'    1042   0631   1034
ASMES     4BF0'    1039   1033
BKMES     4C02'    1048   1035
BLANK1    474A'    0700   0702
BLANKC    4D18'    1204   0195   0390   0423   0698   0750
BLANKS    4736'    0693   0083   0120   0144   0171
BLKTIM    FF28     0041   0193   0205   0206   0209   0389   0401   0402   0405
BREAKC    C500     0035
BS        0800     0031   0232
BUFCTR    FF24     0039   0190   0262   0271   0304   0306   0309   0343   0348
BUFPTR    FF22     0038   0192   0259   0262   0281   0285   0302   0309   0342   0343
CHACNT    FF36     0048   0391   0553   0637
CHOFFS    6000     0011   0178   0181   0244   0250   0554   0596   0624   0650   0654
                          0682
CHWRD     48A2'    0858   0473
CHWRD1    48CC'    0873   0861
CHWRDX    48D8'    0877   0863   0864   0870   0875
COLMAX    0024     0010   0720   0730   0749
CURSOR    4D28'    1212   0217   0413
CWDSR     4018'    0076   0072
CWDSR1    402C'    0083   0166
D10       4D32'    1217   0645
D12       4D30'    1216   0638
D250      4D3A'    1221   0388   0803
D300      4D38'    1220   0382
D40       4D2C'    1214   0266   0290
D47       4D2E'    1215   0830
D5        4D34'    1218   0533
D900      4D36'    1219   0383
DDSAV     FF34     0047   0077   0386   0538   0581
DEBOU1    489A'    0850   0845
DEBOUN    487E'    0839   0425   0448   0497   0588   0849
DECO10    463C'    0590   0586
DECOD1    4604'    0571   0561
DECOD2    4618'    0580   0567   0570   0573
DECOD3    45DE'    0560   0576
DECOD4    465A'    0598   0543
DECOD5    4640'    0591   0584   0589
DECOD7    45FC'    0569   0632
DECOD8    45CE'    0555   0597
DECODE    459C'    0538   0484   0528   0530   0534
DECODX    4650'    0595   0547   0550   0599   0608   0610
DECSP     4666'    0607   0552
DECSP1    4660'    0600   0544
DECSP2    4670'    0611   0601
DECSP3    4686'    0617   0613
DECSP4    4696'    0623   0627
DECSP5    46B4'    0634   0630
DELAY     4372'    0355   0312   0324   0333   0337   0341   0373
DELAY1    4380'    0360   0371
DISPTI    48FE'    0901   0094   0121   0147
DISTON    4D3E'    1223   0145   0335   0395   0444   0499   0592
DSPTI1    4914'    0905   0901
DSPTI2    4926'    0909   0905
DSRLNK    4010'    0072   0069
ENTER     0D00     0033   0229
ENTONE    4D3C'    1222   0331   0419   0450   0493   0764
EOM       4CA2'    1140   0313
ERMES     4BE6'    1036   1035
```

```
LABEL       VALUE   DEFN    REFERENCES

ERRORC      8000    0034    0598
FLGREG      FF26    0040    0191    0213    0380    0409    0427    0482    0483    0531    0565
                            0569    0571    0574    0582    0587    0591    0734    0765    0766
                            0792
FORCE       4D48'   1228    0483    0583    0591    0735    0765
FWD         0900    0032    0234
H0000       4D26'   1211    0088    0775    0798
H02         4D20'   1208    0873
H03         4D1E'   1207    0778    0859
H80         4D2A'   1213    0213    0409
H81         4D1C'   1206    0085
H87         4D22'   1209    0087
HCE         4D24'   1210    0086
HF0         4D1A'   1205    0084    0089
HZ250       4814'   0787    0400    0435    0477    0540    0738    0847
HZ250A      481C'   0791    0788
HZ250B      481E'   0792    0790
HZ250C      484C'   0806    0794
HZ250X      484E'   0807    0797    0799    0802    0805
KDASH       44B4'   0456
KDD         44D4'   0473    0465
KDD0        44DC'   0475    0481
KDD1        44E0'   0477    0500
KDOT        44C8'   0467    0454
KEY         FF95    0061    0103    0130    0156    0220
KEYDN       44FE'   0486    0476
KEYDN0      454E'   0512    0504
KEYDN1      452E'   0502    0498
KEYDN2      4584'   0531    0515
KEYDW0      4464'   0429    0436    0590
KEYDW1      4472'   0434    0432
KEYDW2      4468'   0431    0451
KEYDWN      442E'   0415    0394
KEYSCN      000E    0012    0098    0125    0151    0199
KEYUP       447C'   0437    0430
KEYUP1      4486'   0441
KEYUP2      44AC'   0452    0449
MAXTIM      4D40'   1224    0356    0416    0442    0491    0760
MENS03      40F4'   0151    0154    0165
MENS13      4110'   0160    0158
MENS23      411C'   0164
MENS33      4126'   0170    0161    0163
MENU01      405C'   0098    0101    0107    0109
MENU02      40A8'   0125    0128    0134    0136
MENU1       494A'   0917    0096
MENU11      4960'   0921    0917
MENU12      496A'   0925    0921
MENU13      4980'   0929    0925
MENU14      4996'   0933    0929
MENU15      49AC'   0937    0933
MENU2       49C2'   0945    0123
MENU21      49E4'   0949    0945
MENU22      49EE'   0953    0949
MENU23      4A08'   0957    0953
MENU24      4A22'   0961    0957
MENU25      4A3C'   0965    0961
MENU26      4A56'   0969    0965
MENU27      4A70'   0973    0969
MENU28      4A8A'   0977    0973
MENU3       4AA4'   0985    0149
```

```
LABEL     VALUE   DEFN  REFERENCES

MENU31    4ABC'   0989  0985
MENU32    4AC6'   0993  0989
MENU33    4AE0'   0997  0993
MENU34    4AFC'   1001  0997
MENU35    4B1C'   1005  1001
MENU36    4B3E'   1009  1005
MENUS1    404C'   0094
MENUS2    4094'   0120
MENUS3    40D6'   0144  0223  0398  0433
MESDS1    4722'   0681  0685
MESDSP    470E'   0672  0095  0097  0122  0124  0148  0150  0173  0687
MSEC10    4D42'   1225  0844
OFFSET    FF66    0054  0513  0862  0865  0871  0876
OLDIN     FF38    0049  0798  0807
PAD       83E0    0036  0037  0038  0039  0040  0041  0042  0043  0044  0045
                        0046  0047  0048  0049  0050  0051  0052  0053  0054
                        0055  0056  0057  0058  0059  0060  0061  0062  0063
                        0888
PGTMAX    0600    0009  0194  0307
PLAYER    FF94    0060  0088
R0        0000          0245  0250  0254  0274  0281  0301  0345  0370  0386
                        0453  0458  0461  0464  0469  0472  0538  0562  0581
                        0614  0634  0639  0640  0643  0644  0645  0649  0650
                        0651  0652  0672  0686  0742  0885  0890
R1        0001          0100  0102  0103  0104  0106  0108  0110  0111  0112
                        0112  0113  0114  0115  0127  0129  0130  0131  0133
                        0135  0137  0138  0139  0153  0155  0156  0157  0160
                        0162  0164  0170  0174  0175  0176  0203  0206  0207
                        0210  0211  0224  0225  0228  0229  0232  0234  0236
                        0238  0240  0248  0251  0252  0255  0264  0266  0275
                        0277  0288  0290  0299  0300  0301  0304  0307  0311
                        0323  0328  0330  0336  0340  0346  0372  0402  0403
                        0406  0407  0420  0421  0453  0460  0462  0463  0464
                        0468  0470  0471  0472  0478  0479  0480  0503  0505
                        0507  0508  0509  0514  0518  0519  0520  0521  0535
                        0536  0555  0556  0563  0615  0617  0618  0620  0621
                        0626  0635  0637  0638  0643  0646  0647  0662  0663
                        0664  0665  0673  0676  0677  0678  0679  0680  0693
                        0694  0695  0696  0697  0699  0701  0712  0713  0724
                        0725  0726  0727  0740  0825  0831  0887  0888  0889
                        0890
R10       000A          0358  0368  0370  0438  0440  0446  0487  0489  0495
                        0611  0620  0623  0628  0629  0631  0827  0829  0840
                        0842  0844  0846  0848
R11       000B          0737  0739  0846  0848
R12       000C          0079  0080  0146  0146  0303  0355  0359  0374  0392
                        0399  0415  0424  0434  0437  0441  0474  0486  0490
                        0539  0759  0767  0774  0839  0850  0858  0895
R15       000F          0084  0085  0086  0087  0179  0182  0215  0217  0227
                        0254  0258  0279  0315  0411  0413  0423  0558  0625
                        0652  0656  0663  0665  0678  0680  0683  0695  0697
                        0700  0715  0717  0721  0725  0727  0731  0746  0748
                        0751
R2        0002          0175  0178  0179  0180  0181  0182  0219  0220  0221
                        0228  0240  0242  0243  0244  0245  0246  0247  0248
                        0257  0258  0265  0267  0289  0291  0461  0462  0469
                        0470  0506  0507  0512  0513  0514  0517  0519  0522
                        0523  0524  0525  0526  0527  0529  0548  0549  0551
                        0554  0558  0595  0596  0600  0607  0609  0611  0642
                        0646  0653  0654  0655  0656  0674  0675  0684  0711
                        0712  0714  0715  0716  0717  0741  0744  0745  0746
                        0747  0748  0754  0755  0886  0891
R3        0003          0318  0319  0321  0338  0458  0459  0460  0524  0526
                        0623  0624  0625  0640  0641  0642  0681  0682  0683
                        0718  0719  0721  0728  0729  0731  0750  0751  0889
                        0893
R4        0004          0076  0077  0078  0079  0088  0089  0099  0100  0103
                        0126  0127  0130  0139  0152  0153  0156  0174  0190
                        0191  0192  0193  0195  0197  0198  0200  0201  0202
                        0203  0205  0206  0209  0213  0215  0220  0227  0243
                        0259  0262  0262  0271  0279  0281  0285  0299  0302
                        0303  0304  0306  0306  0309  0309  0342  0343  0343
                        0348  0348  0374  0380  0381  0382  0383  0386  0387
                        0388  0389  0390  0391  0392  0399  0401  0402  0405
                        0409  0411  0418  0424  0427  0428  0434  0447  0474
                        0478  0480  0482  0483  0496  0503  0506  0509  0510
                        0512  0513  0516  0517  0521  0522  0527  0529  0531
                        0532  0533  0536  0538  0539  0553  0565  0569  0571
                        0574  0581  0582  0587  0591  0637  0639  0719  0729
                        0734  0762  0763  0765  0766  0767  0792  0798  0800
                        0801  0803  0804  0807  0821  0827  0832  0833  0850
                        0862  0865  0867  0871  0876  0895
R5        0005          0094  0096  0121  0123  0147  0149  0172  0194  0197
                        0201  0255  0260  0286  0313  0315  0316  0317  0317
                        0318  0320  0326  0672  0673  0674  0681  0686  0698
                        0700
R6        0006          0170  0183  0196  0198  0202  0210  0224  0251  0261
                        0265  0269  0272  0277  0283  0287  0289  0293  0385
                        0406  0420  0555  0559  0560  0568  0575  0612  0617
                        0621  0622  0755
R7        0007          0384  0456  0457  0467  0541  0542  0542  0545  0546
                        0549  0580  0598  0720  0722  0730  0732  0734  0735
                        0737  0739  0749  0752  0867  0868  0869
R8        0008          0565  0566  0571  0572  0582  0583  0585  0789  0791
                        0792  0793  0795  0796  0796  0806  0807  0821  0822
                        0823  0825  0865  0866  0869
R9        0009          0445  0446  0447  0494  0495  0496  0828  0830  0831
RAMBUF    FF40    0053  0718  0728
RBKMAX    0700    0020  0403
RDBIT     0000    0007
RECLIN    000A    0021  0562  0614  0634
RECMAX    0073    0025  0546
RECTAB    4CA4'   1145  0549
RECV      43A2'   0380  0185  0657
RECV0     43DA'   0392  0511
RECV00    43DE'   0393  0404  0412  0414  0593
RECV2     43F2'   0399  0397
RECV3     4426'   0413  0410
RECV4     43E2'   0395  0426
RESET     410C'   0159  0105  0132
RESET0    48DA'   0885  0159
RESET1    48EC'   0890  0892
REIURN    48F4'   0895  0885
RR6MAX    03BE    0024  0560
RR6MIN    039A    0023  0385  0612
RSCRNO    0232    0022  0563  0615  0635
RSTADR    0012    0002  0897
RTFLAG    4D46'   1227  0427  0793
RWPM      01F9    0019  0647
SAVACC    FF6E    0058
```

```
LABEL     VALUE    DEFN  REFERENCES

SAVR12    FF2C     0043  0079  0146  0303  0374  0392  0399  0424  0434  0474
                         0539  0767  0850  0895
SAVR5     FE3A     0050  0197  0201
SAVR6     FF3C     0051  0198  0202
SAVVDP    FFF4     0063  0089
SBFCTR    FF3E     0052  0078  0306  0348
SCALE     4854'    0821  0452  0502
SCALE1    4864'    0827  0824
SCALE2    4874'    0832  0826
SCNMAX    03C0     0008  0699
SCROL1    4770'    0721  0723
SCROL2    478A'    0731  0733
SCROL3    47D4'    0757  0736
SCROL4    479C'    0737  0758  0768
SCROL6    475E'    0714  0743
SCROL8    47C4'    0751  0753
SCROLL    4754'    0711  0276  0347  0564  0616  0636
SEC       FF30     0045  0387  0639  0804
SEMIFR    4D44'    1226  0585  0587  0766
SGCADR    8400     0001  0113  0115  0145  0331  0335  0395  0419  0444  0450
                         0493  0499  0592  0764
SHC       4800'    0774  0396  0431
SHCX      4812'    0780  0777
SHIFTC    0200     0030  0221
SHIFTM    3B00     0027  0164
SHIFTQ    0500     0026  0104  0131  0157
SHIFTR    0600     0029  0162
SHIFTT    0700     0028  0160  0183
SKMES     4BFC'    1045  0629  1034
SPMES     4BDA'    1033  0611
STATUS    FF9C     0062  0100  0127  0153  0203
STOCUR    FF20     0037  0195  0215  0227  0243  0279  0390  0411  0887
SUBSEC    FF2E     0044  0388  0801  0803
SUM901    FF70     0059  0428  0447  0496  0763  0827  0832
TONFRQ    4C08'    1055  0112
TWOEOC    FF6C     0057  0383  0478  0512  0517  0521  0522  0536  0867
TWOSPA    FF6A     0056  0382  0503  0506  0509  0527  0529
VDPWAD    4704'    0662  0177  0212  0226  0253  0256  0278  0308  0408  0422
                         0557  0619  0648
VRD       FBFE     0005  0279  0315  0721
VWA       8C02     0003  0004  0005
VWD       FFFE     0004  0179  0182  0215  0217  0227  0254  0258  0411  0413
                         0423  0558  0625  0652  0656  0683  0700  0731  0751
WDBIT     4000     0006  0176  0194  0211  0225  0252  0407  0421  0556  0618
                         0647  0676  0693  0713  0744  0754
WORD      4D4A'    1229  0482  0531  0566  0569  0572  0574
WORDC     FF68     0055  0381  0510  0516  0532  0533
XBKMAX    0070     0014  0207
XMIT      4160'    0190  0184  0349
XMIT0     417E'    0197  0208  0216  0218  0280
XMIT00    41CC'    0218  0237  0239  0249
XMIT2     41C6'    0217  0214
XMIT3     41DE'    0224  0222
XMIT4     41CE'    0219  0204
XMIT5     41FA'    0232  0230
XMIT6     4220'    0245  0241
XMIT7     4270'    0271  0263
XMIT8     4286'    0277  0268  0270  0273  0282  0284  0292  0294
XMTAB     4C10'    1067  0175  0301
XMTBS     4296'    0281  0233
```

```
LABEL     VALUE   DEFN  REFERENCES

XMTDD     4C2C'   1081  0248  0317
XMTEN     42BE'   0299  0231
XMTEN1    42E4'   0309  0344
XMTEN2    42FA'   0315  0310
XMTEN3    4304'   0318  0314
XMTEN4    431A'   0326  0322  0339
XMTEN5    4324'   0330  0327
XMTEN6    4328'   0331  0329
XMTEN7    4350'   0342  0325
XMTEN8    42DC'   0307  0305
XMTFWD    4216'   0242  0235
XMTLIN    0009    0015  0274  0345
XMTRE1    4B84'   1021  1017
XMTRE2    4BB2'   1025  1021
XMTREC    4B5C'   1017  0172
XMTSEL    FF2A    0042  0139  0174  0299
XR6MAX    01B6    0018  0272
XR6MIN    0192    0017  0196
XSCRNO    0052    0016  0275  0283  0346
XWPM      0019    0013  0176
```